e-Business & Contacts center Solutions

Hermes.Net 4

VOCALCOM

# Interface designer

## Manuel Utilisateur

## Hermès.Net Version 4.0

Call Center Technology

## Summary

**Summary** ..... 2
1. Introduction ..... 5
1.1 Glossary ..... 5
2. Script Manager ..... 6
2.1 Header ..... 6
2.2 Status bar ..... 6
2.3 Scripts list ..... 7
2.3.1 Filters and commands on the script list ..... 7
2.4 Script list functionalities description ..... 7
2.4.1 Script groups ..... 9
2.4.2 Script list ..... 9
2.5 Création de nouveaux scripts. ..... 10
2.6 Importation de scripts. ..... 10
3. Interface Designer ..... 11
3.1 Présentation de l'interface ..... 11
3.2 Les menus ..... 12
3.2.1 Script menu ..... 13
3.2.2 Pages Menu ..... 17
3.2.3 Menu Edition ..... 19
3.2.4 Template Menu ..... 22
3.2.5 Menu Connections ..... 22
3.2.6 Menu Modules ..... 23
3.2.7 Preferences Menu ..... 24
3.2.8 Tools Menu ..... 25
3.3 The menu bar ..... 26
3.3.1 Move menus in the bar. ..... 26
3.3.2 Removing menus in the bar. ..... 26
3.3.3 Flotting Menus. ..... 26
3.3.4 The 3 types of menus representation. ..... 27
3.3.5 Customizing the menu bar. ..... 27
3.3.6 Particular case of the item menu. ..... 28
3.4 The available objects ..... 29
3.4.1 Summary ..... 29
3.4.2 Notation, and use of objects specificities ..... 31
3.5 Properties Tabs ..... 32
3.5.1 Use. ..... 32
3.5.2 The different modes of representation. ..... 32
3.5.3 The different properties tabs. ..... 33
3.6 Status bar ..... 50
4. « CTI Events» Appendix ..... 51
4.1 CTI event principal ..... 51
4.2 List of events sent to the script by the workspace ..... 52
4.2.1 OnBeforeOpenStatus ..... 52
4.2.2 OnStatus ..... 52
4.2.3 CallSuccess ..... 53
4.2.4 CallPreview ..... 53
4.2.5 CallFailure ..... 53
4.2.6 ConsultFailure ..... 53
4.2.7 OnBeforeConsult ..... 53
4.2.8 OnConsult ..... 53
4.2.9 QueueStateChange ..... 54
4.2.10 CallAlerting ..... 54
4.2.11 CallCleared ..... 54
4.2.12 CallTransfer ..... 54
4.2.13 OnBeforeTransfert ..... 54
4.2.14 OnTransfert ..... 54
4.2.15 OnInfo ..... 55


Vocalcom - 7 rue de Tilsitt – 75017 Paris - Tél. : +33 1 55 37 30 50 – Fax : +33 1 55 37 30 51 – www.vocalcom.com
SA au capital de 5 000 000 € – RCS Paris B 401 973 631 – TVA Intracommunautaire : FR 27401973631

4.2.16 OnBeforePause ... 55
4.2.17 OnPause ... 55
4.2.18 OnPauseEnd ... 55
4.2.19 SearchModeStateChange ... 55
4.2.20 OnRecordStart ... 55
4.2.21 OnEndRecord ... 55
4.2.22 OnHold ... 55
4.2.23 OnHoldEnd ... 56
4.2.24 OnReceiveContactInfos ... 56
5. Appendix « Creating global variables » ... 57
5.1 Basic principle ... 57
5.2 Free variables ... 57
5.3 Connected variables ... 58
5.4 Dynamic Variables ... 58
5.5 Auto creation of variables. ... 59
5.6 Utilisation des variables. ... 59
6. Appendix « Connections and management of customer tables» ... 61
6.1 Setting the parameters of auto-connection ... 62
6.2 Setting the auto-connection filter ... 63
6.3 Explanation of the menu "Actions" ... 64
6.4 Management of objects to connect ... 65
6.4.1 General interface ... 65
6.4.2 Details on connection indicators reading and writing ... 66
6.4.3 Details on the manual editing of connections ... 67
6.5 Registration Window ... 68
6.6 Window limiting the number of pages ... 69
6.7 Examples of uses ... 70
6.7.1 Using similar to the application for connection to the client file Scripter v3. ... 70
6.7.2 Disconnecting all the fields in the script ... 71
6.7.3 Modification of all existing connections of the script ... 71
7. Annex "Checking connections" ... 72
8. Appendix "Query Editor" ... 73
8.1 Basics ... 73
8.1.1 Elements list of the query editor ... 75
8.1.2 Window parameter of the request ... 76
8.2 Select query ... 76
8.3 Query insert from another table ... 78
8.4 Insert query ... 79
8.5 Request update ... 80
8.6 Request delete ... 81
8.7 Creating a join between two tables ... 81
8.8 Creation of advanced criteria ... 83
8.9 Executing the query ... 84
8.10 Frequent Error Messages when editing existing queries ... 84
9. Annex « Action Builder " ... 86
9.1 The interface ... 86
9.1.1 Menus. Provides access to various functions of the editor. ... 87
9.1.2 Orders editions. Allows to place on the working area of the navigation commands, testing, and verification. 87
9.1.3 Orders of shares. Puts in the work zone orders. ... 87
9.1.4 Displaying the name of the action. The disk icon allows you to save the action. ... 87
9.1.5 display area from the list of commands present in the action. Select a command from the list to display the settings for it. ... 87
9.1.6 Viewing and editing the name of the command. Each order can be customized with a specific command name. Under the command name is the type of command. ... 87
9.1.7 Viewing and editing the list of parameters corresponding to the selected action. ... 87
9.1.8 Display of various types of global variables available depending on the command. ... 87
9.1.9 Work area. Commands selected in the left menu are placed in this area. ... 87
9.2 Les menus ... 88


Vocalcom - 7 rue de Tilsitt – 75017 Paris - Tél. : +33 1 55 37 30 50 – Fax : +33 1 55 37 30 51 – www.vocalcom.com
SA au capital de 5 000 000 € – RCS Paris B 401 973 631 – TVA Intracommunautaire : FR 27401973631

9.2.1 Menu Actions ..... 88
9.2.2 Menu Preference ..... 89
9.2.3 Menu Tools ..... 89
9.3 Use of the interface ..... 90
9.3.1 Ajout de commandes ..... 90
9.3.2 Suppression de commandes ..... 91
9.3.3 Removing links between commands. ..... 91
9.3.4 Illegal connections. ..... 91
9.3.5 Various edition commands. ..... 92
9.4 Les commandes ..... 93
9.4.1 Commands for editing and testing. ..... 93
9.4.2 Main action commands. ..... 94
9.4.3 Commandes avancées. ..... 98
9.4.4 Database commands. ..... 100
9.4.5 Navigation commands. ..... 103
9.4.6 CTI Commands ..... 105
9.4.7 Advanced CTI Commands ..... 106
9.4.8 CRM Contact Commands. ..... 109
9.4.9 Information on command CRM Contact. ..... 110
9.4.10 V.M.C Commands ..... 111
9.5 action Debugger ..... 113
9.5.1 Principle ..... 113
9.5.2 Interface ..... 113
9.5.3 Example debug action ..... 114
10. Appendix « Diagram Viewer » ..... 117
10.1 Description ..... 117
11. Appendix « Page template » ..... 118
11.1 Création ..... 118
11.2 Affectation ..... 119
12. Appendix « Search mode » ..... 120
13. Appendix « criteria editor » ..... 121
14. Appendix « Script Debugger» ..... 122
15. Aappendix « create your own style sheet » ..... 124
15.1 Import CSS files ..... 124
15.2 Select a style to apply to script ..... 124
15.3 Object structure in DOM and names of the classes used ..... 125
15.3.1 Object « label » ..... 125
15.3.2 Object « panel » ..... 125
15.3.3 Object « text field » ..... 126
15.3.4 Objet « date field » ..... 126
15.3.5 Object « Time picker » ..... 127
15.3.6 Objet « Text area » ..... 127
15.3.7 Object « combo » ..... 127
15.3.8 Object « list box» ..... 128
15.3.9 Objet « button » ..... 128
15.3.10 Object « radio button» ..... 128
15.3.11 Objet “Check box” ..... 128
15.3.12 Objet « Hyperlink » ..... 128
15.3.13 Object « data grid » ..... 129
15.3.14 Object « objection » ..... 129
15.3.15 Object « IFrame » ..... 130
15.3.16 Object « credit card » ..... 130
15.4 Using images in the CSS files ..... 131
15.5 Using tifferent style sheets in the same script ..... 131
15.6 Appendix « replace database connections during script import » ..... 132

# 1. Introduction

The application « Interface Designer » of the Hermes.Net V4 solution allows you to create and manage your screen scripts, your agent interfaces as well as your interactive voice scripts.

For each of these types of scripts, a specialized interface is proposed. This Manual will describe in detail all the available functionalities of the script designer interface.

For the other script types, please refer to the following manuals:

For IVR: manual “Interactive voice recognition”
For the agent interface: manual “Workspace editor”

The main functionalities described in this manual are:

- Scripts management
- Edition functions
- Actions management
- Client files management
- Style sheets management
- Scripts generation
- Import / Export of scripts

## 1.1 Glossary

- **Script objects**: defines an object placed on a page of the script. This object can be read only (Label, Image, Panel, etc) or in read and write (Text field, Datagrid, etc). Each page object has a name (or id), this name is used to refer the object in the actions (see later in this document). The prefix used to obtain the properties or field methods is “@”.
  Ex: @MyTextBox.
- **Global Variable**: variables are created using the editor or passed on using the launching url and are available throughout the script. The prefix used to reach a global variable is: “$”.
  Ex: $MyGlbVar.
- **Action**: collection of simple commands created using the “Action builder” module which can be triggered on different script events.

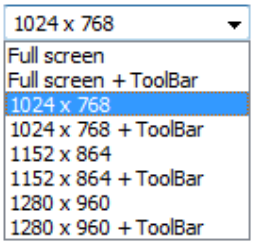

Vocalcom - 7 rue de Tilsitt – 75017 Paris - Tél. : +33 1 55 37 30 50 – Fax : +33 1 55 37 30 51 – www.vocalcom.com
SA au capital de 5 000 000 € – RCS Paris B 401 973 631 – TVA Intracommunautaire : FR 27401973631

# 2. Script Manager

The home screen of the interface allows managing the existing scripts, import or create new ones. It consists of 3 main areas, a header and a status bar.
Each of the main areas is accessible by clicking the corresponding tab.

## 2.1 Header

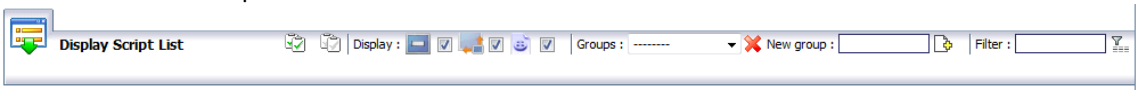
The only available information in this area is the screen resolution in which the editor will be launched. The possible resolutions are:




For each of these choices, it is possible to display the address bar of the navigator.

The chosen resolution is saved automatically for each individual user of the Interface Designer modules.

**Part 1** : Scripts list




**Part 2** : Creation of new scripts

**Part 3** : Import new scripts

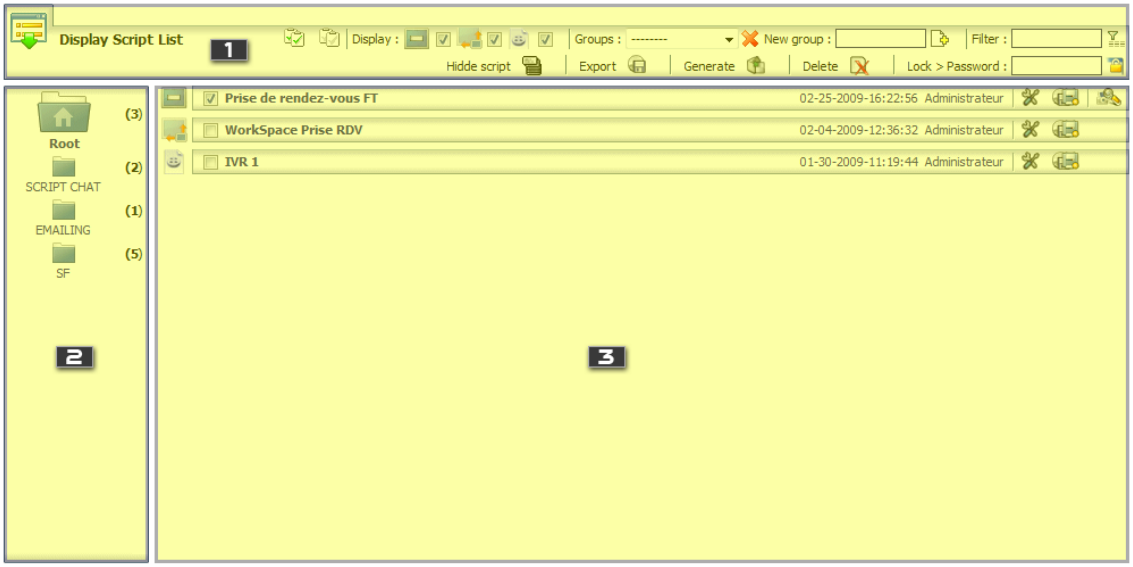
## 2.2 Status bar

Allows :

Quitting the application (“Exit”)
To see information concerning your login, company and identifier.
The application version. The version could be asked by the support helpdesk.

## 2.3 Scripts list

The « Script list » has made of 3 areas:




### 2.3.1 Filters and commands on the script list

This zone has an always visible top area, and a lower area only visible if at least one script is selected.

## 2.4 Script list functionalities description

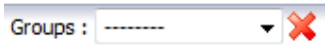
Allows selecting / unselecting the scripts from the selected group.

Allows filtering the script list according to the selected script

Screen

Agent Interface

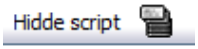
IVR



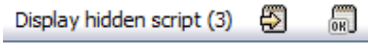

Displays the list of existing groups. Allows deleting the group. The scripts belonging to the deleted group will automatically by brought back to the root.

Creation of a new script group

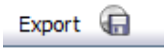
Allows filtering the list of script by their names. A wildcard (*) can be used.



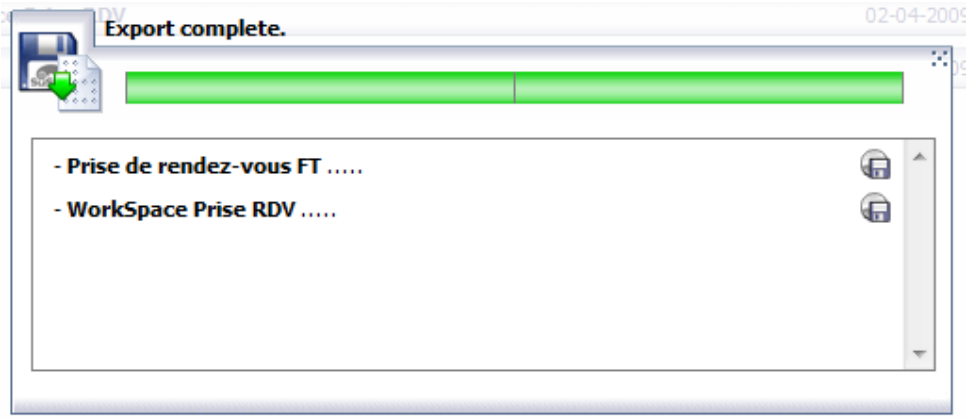

This command allows hidding the selected scripts. The hidden scripts are removed from the list and can be displayed at any moment by selecting one of the following commands.



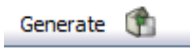

Allows displaying scripts that were hidden. The value in parenthesis indicates how many scripts are hidden. The first icon allows restoring all scripts, the second, displaying the list of hidden scripts and restoring them individually.



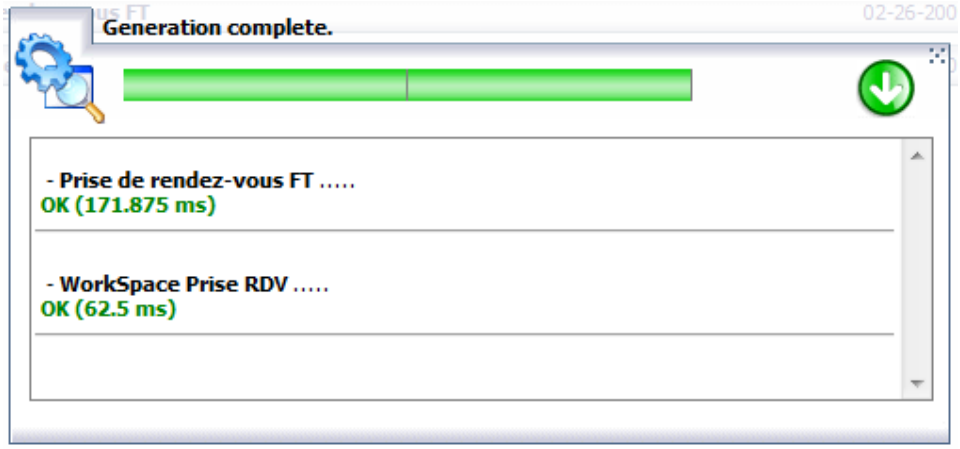

Allows exporting the selected and get the .WSS (Web Scripter Script) files. The following window appears when the export starts.



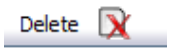

Click on the floppy icon to download the .WSS file.



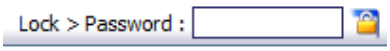

Allows generating (Publish) the selected scripts. When the generation starts, the following windows appears.




The generation progress appears in green. Red or orange indicates error scripts.



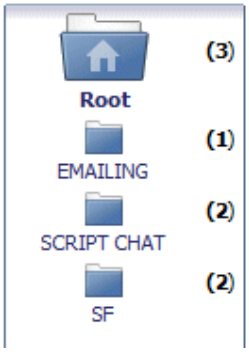

Used to delete the selected script(s). It will be asked to confirm.




Allows locking the selected scripts. When a script is locked, it's impossible to edit it, generate it, export it or delete it. All these operations will be available again when the script is unlocked.

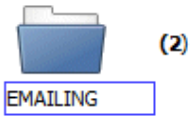
### 2.4.1 Script groups

The script group allows organising the scripts.




The counter on the right of the group indicates the amount of scripts present in the group. The group « Root » indicates the default group (in other words : no group).

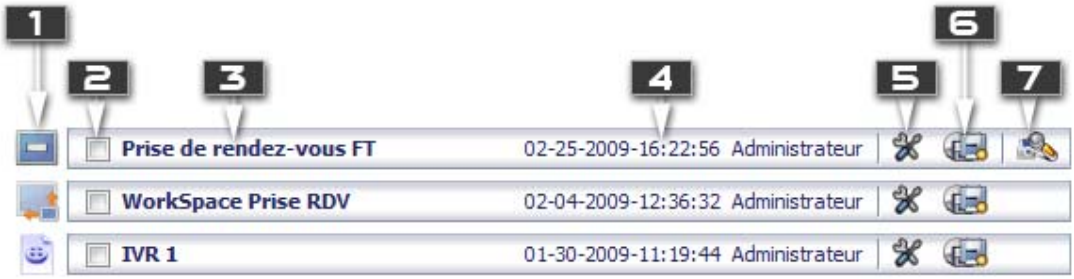
It is possible to rename a group (except “root”) by double clicking on the name, a field appears with the name of the actual group:




Type in the new name and press enter.

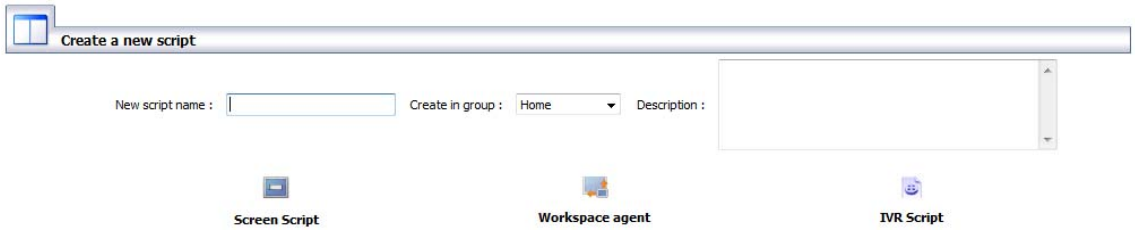
**Add a script to a group.**
You can add or remove a script from a group using « drag and drop » with your mouse. Click on the script icon to move and drop it in the preferred group.

### 2.4.2 Script list



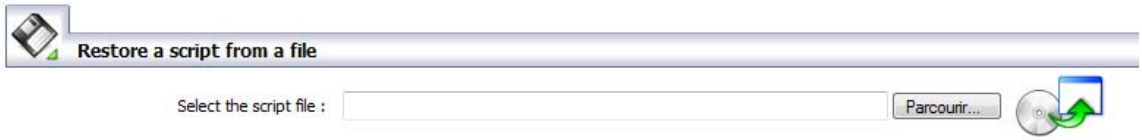

**1a –** double click on the icon to open the script using the appropriate editor (Interface Designer, WorkSpace Designer, IVR Editor).
**1b -** Drag & drop this icon to move the group to the selected group.
**2 -** Check this box to select the script and access to the various functions described above (2.1.1).
**3 –** Script name.
**4 -** Date, time of last modification. Name of the user who made the change.
**5 -** Same functionality as the double click on the icon of the beginning of the line.
**6 –** Click this icon to duplicate the script.
**7 –** Click this icon to view the published script. Only scripts generated on mode "production" are pre-viewable.

## 2.5 Création de nouveaux scripts.




This tab allows you to create a new script based on the icon clicked (Script screen, interface agent, IVR).
Enter the name of the new script, choose from the dropdown list the group in which to create this script, and enter a description (optional).
Click the icon for the type of script you want.

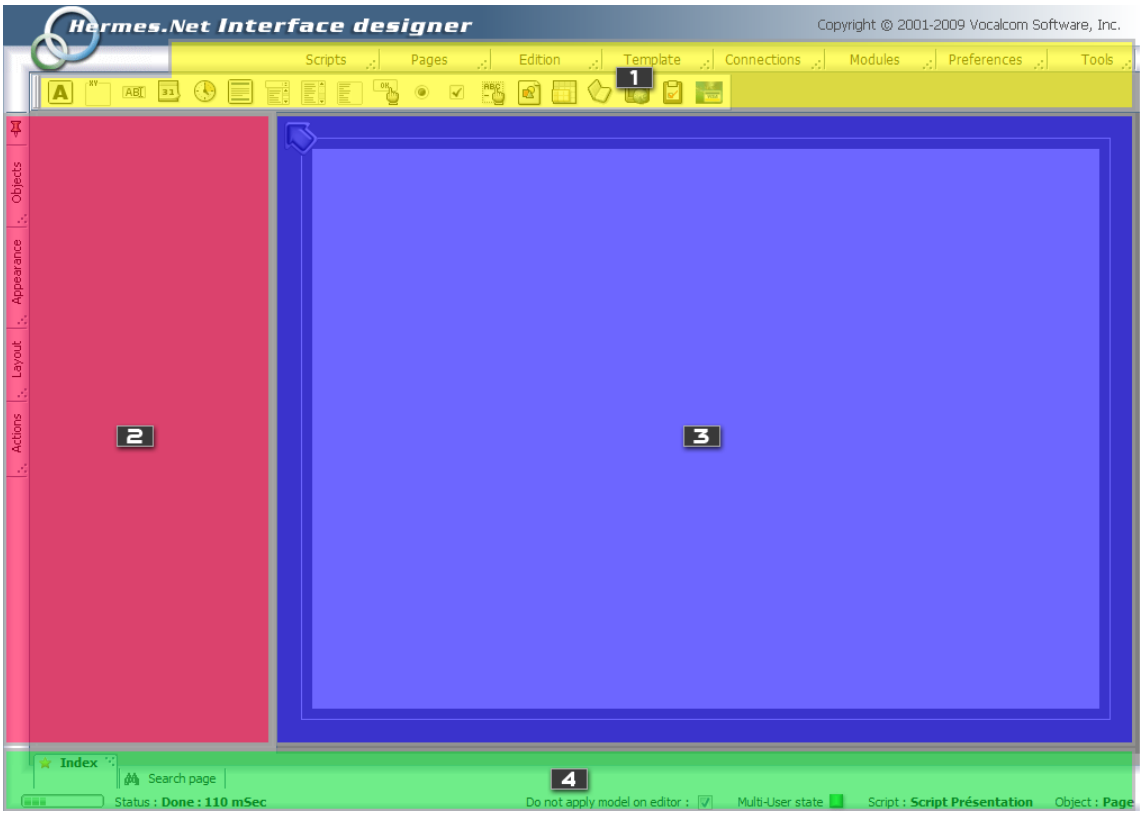
## 2.6 Importation de scripts.




This tab allows you to import a script saved. The file extension must be imported. WSS, the type of script is included in the format of the WSS.
Choose the disk on your WSS file, then click on the icon to the right.

# 3. Interface Designer

## 3.1 Présentation de l'interface




1 **Menus.** Provides access to various actions of the interface.

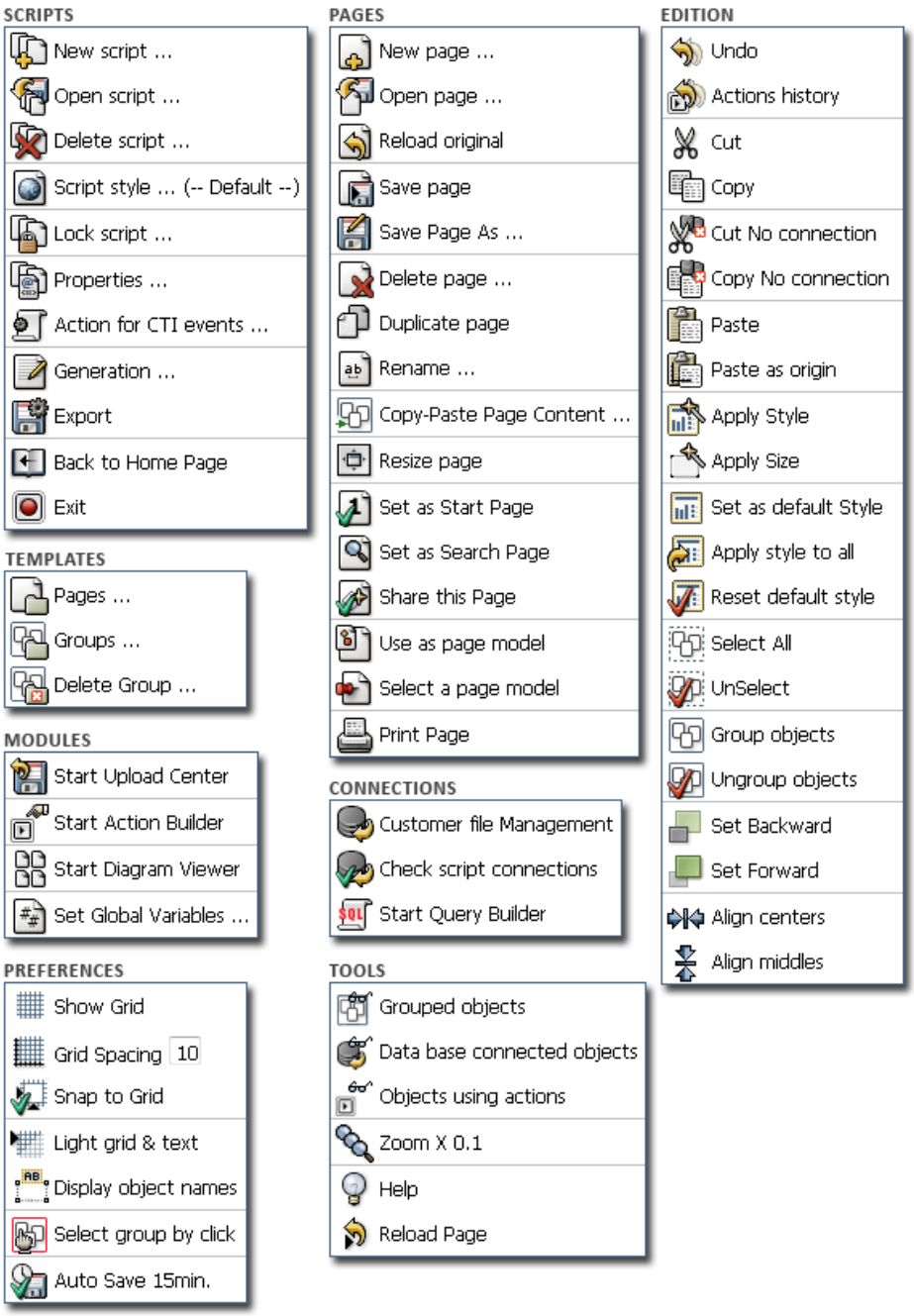
2 **Properties.** Lists tab allows access to different groups of objects and properties of the page.

3 **Script page.** Display area of the current page.

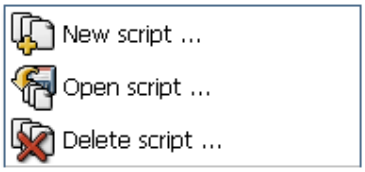
4 **Pages list / Status / various infos**

## 3.2 Les menus

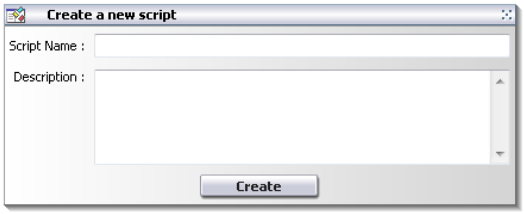
The following screens represent all active items. Activation of items depends on the current state of the interface .



### 3.2.1 Script menu

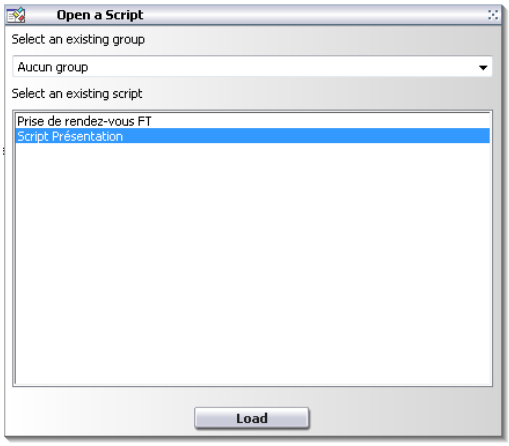



**New Script ...**



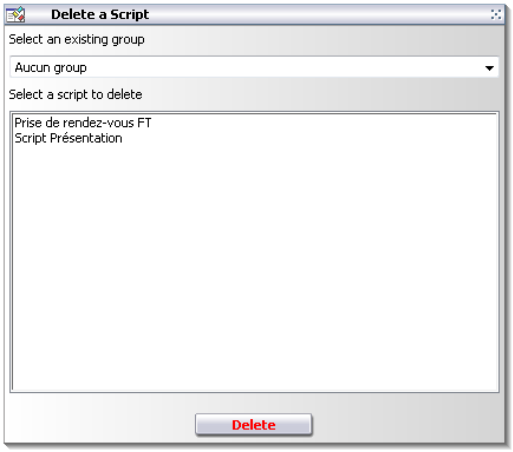

You can create a new script. Enter a name and description (optional).

**Open Script ...**



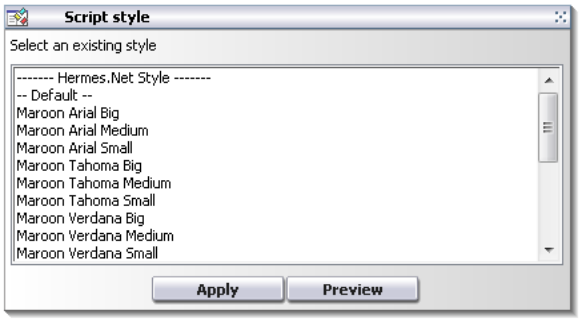

Select the group where the script is placed, then click "Load".

**Delete Script ...**



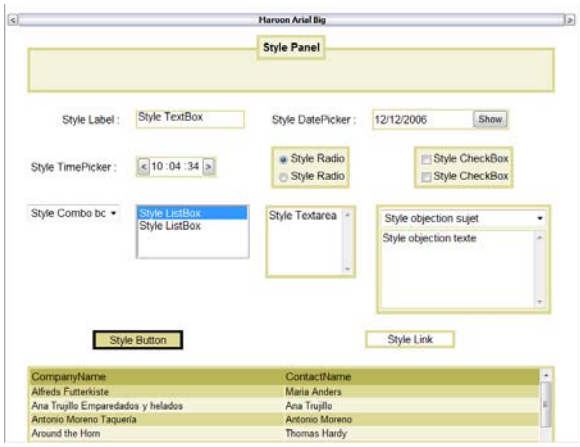

Select the group where the script to be deleted and click "Delete." A confirmation is requested.
You can select multiple scripts to be deleted in one command. For this, use the CTRL key, click on the names of scripts to delete.

**Script Style ...**




The script interface screen allows you to choose a style to apply to your pages. Several default styles are offered.
The style list is composed of two parts, a "default" container styles provided with the interface and a "custom" that contains the styles created by the user. See Schedule "Creating your own style".

You can view the styles by clicking the "Preview", the following screen appears :




Using the left and right arrows you can scroll through the available styles. When you close this screen, the list goes back on the latest style displayed.

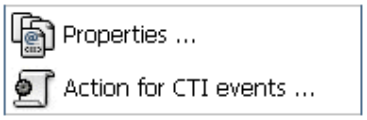
---

**Lock Script ...**

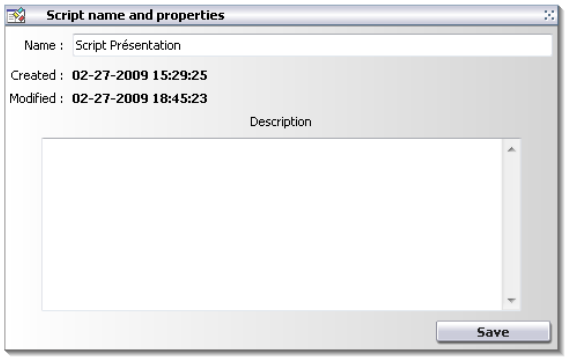
To lock a script with a password. It is possible to lock the script or just lock the script and exit.

---


Vocalcom - 7 rue de Tilsitt – 75017 Paris - Tél. : +33 1 55 37 30 50 – Fax : +33 1 55 37 30 51 – www.vocalcom.com
SA au capital de 5 000 000 € – RCS Paris B 401 973 631 – TVA Intracommunautaire : FR 27401973631




**Properties…**

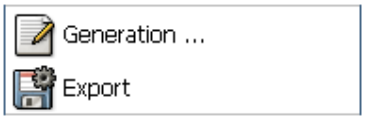



Allows you to rename a script, and add a description.

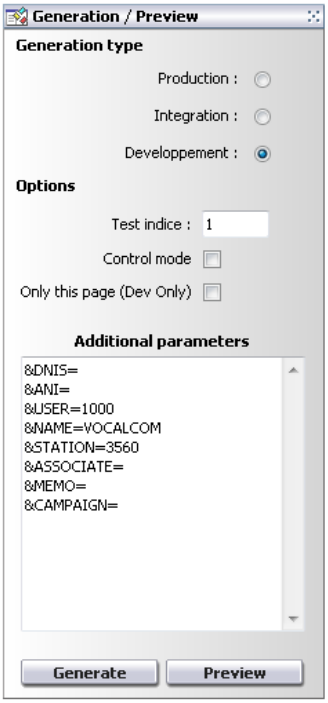
**Action for CTI events…**

Can associate a script action with a CTI agent event. The part "CTI Events" describes how information is exchanged.


Vocalcom - 7 rue de Tilsitt – 75017 Paris - Tél. : +33 1 55 37 30 50 – Fax : +33 1 55 37 30 51 – www.vocalcom.com
SA au capital de 5 000 000 € – RCS Paris B 401 973 631 – TVA Intracommunautaire : FR 27401973631




**Generation…**

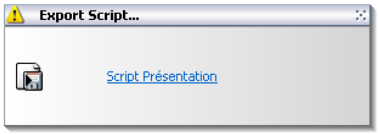



This screen allows you to generate the script on the publication platform defined in the administration.
A radio box lets you choose what kind of generation to do: Production, Integration, Development. Note that only the scripts generated in production mode will be visible from the administrator.

The options allow:
- To choose the value of the test index. The fields of the script connected to a customer table will be displayed on the index chosen.
- To choose whether the script debugger must be displayed (see Appendix "script debugger).
- To choose whether only the current page must be generated.

Additional parameters can be modified. Note that it is possible to add their own settings, and that they will be converted automatically into global variables.

**Export…**




Allows exporting the scripts into a .WSS file

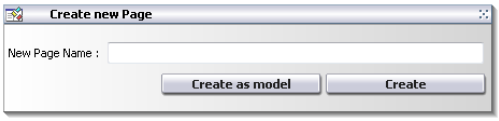
---

**Back to home page**
Goes back to « Script Manager »

**Exit**
Exits application and closes the browser

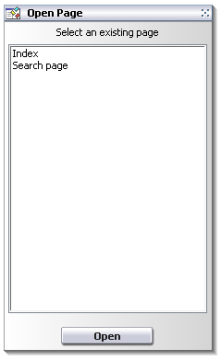
### 3.2.2 Pages Menu

**New Page …**




Create a new page. Click on "Create as model" if you want to create this page as a page template. See Schedule "pages template"

**Open Page …**




Displays a list of pages. Select the page to open and click "Open."

**Reload original**
Allows you to retrieve the page as such she was saved for the last time.

---

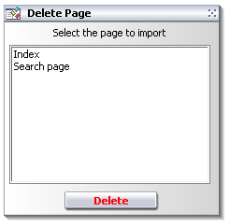
**Save Page**
Saves the current page

**Save Page As …**

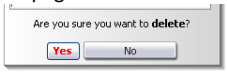
Sauvegarde la page sous un autre nom.

---

**Delete page …**



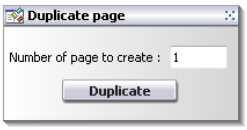

Select the appropriate pages to remove and then click "Delete."




Click « yes » to confirm the action.

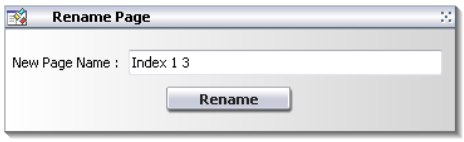
: To select multiple pages, use the mouse and click CTRL key on the keyboard simultaneously.

**Duplicate page …**




Enter the number of page identical to the current page to create, then click "Duplicate". The name of the created pages are indexed. Ex: Index duplicated 3 times gives pages Index 1, Index 2, Index 3.

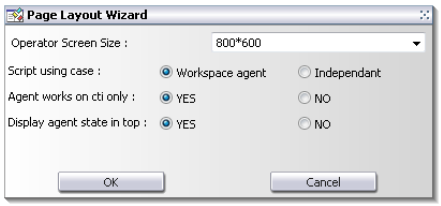
**Rename …**




Allows renaming a page

**Copy-Paste Page Content …**
Aimed at advanced users.
Allows you to change the source code of the page.

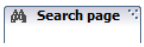
**Resize page …**


Page Layout Wizard
Operator Screen Size : 800*600
Script using case : Workspace agent Independant
Agent works on cti only : YES NO
Display agent state in top : YES NO
OK
Cancel

This menu allows you to calculate the size of the page depending on the settings screen agent and configuration of its WorkSpace.

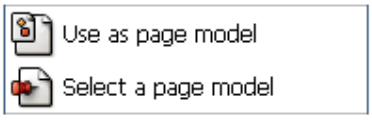
**Set as Start Page**
Permet de marquer cette page comme page de démarrage du script. La page de démarrage est repérée dans les onglets de page grâce à l'icône suivant :

**Set as Search Page**
Allows you to mark this page as a search page. See Appendix "Search Mode"

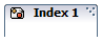



**Share this Page**
By checking this option, you allow other users of your company to load your page from the menu "Template - Import page".

: The sharing of a page does not cause the pooling of the page. The changes made by other users are not reflected on your page.




**Use as page model**
By clicking on the Item page in the course becomes a page template, it could be modeled on another page. The template pages of script are marked as follows in the tab page.




The check box is placed on the bottom bar of the editor to display or not the model on the current page.

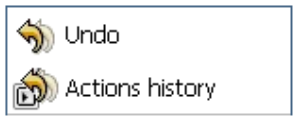
**Select a page model**
Can assign a page template to the current page or any pages of the script.

Choose the model from the list and click "Apply." Check the box "Apply to all model page" if you want this model to all pages of the script.
Click "Remove" if you no longer want to use as a model for the current page.

---

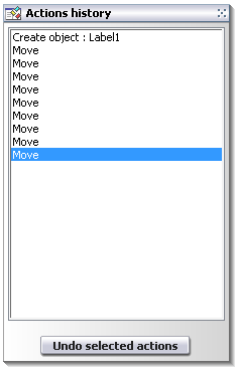
**Print Page …**
Displays the print preview of the current page.

### 3.2.3 Menu Edition




**Undo (CTRL-Z)**
Cancels all previous actions
A maximum of 10 actions can be canceled through the CTRL-Z.

**Actions history.**

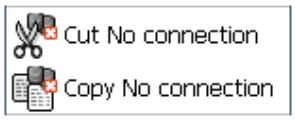



Allows you to view a list of 10 (maximum) past actions and choose which to cancel.

---

**Cut (CTRL-X)**
Deletes the object and places it in the clipboard.
**Copy (CTRL-C)**
Copies the selected object to the clipboard.

---




**Cut No connection**
Same as the previous command, but the object placed in the clipboard does not connect to the database if it has one assigned.
**Copy No connection**
Same as the previous command, but the object placed in the clipboard does not connect to the database if it has one assigned.

---

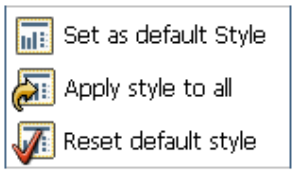
**Paste (CTRL-V)**
Inserts the contents of the clipboard into the current page. Place the pasted object over the object copied.
**Paste as origin**
Inserts the contents of the clipboard into the current page. Place the object stuck in the same location as the copied object.

The objects of the clipboard can be copied from one page to another.

---

**Apply Style**
Allows you to use only the style of the copied object to the selected object**.**
**Apply Size**
Allows you to use only the size of the object copied to the selected object.

---

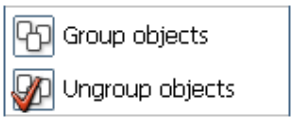



**Set as default Style**
After selecting an item, click on this item. All objects created thereafter will have the same style as the original object.
**Apply style to all**
Can assign the default style to all objects of the same type of page.
**Reset defaut style**
Reset default style

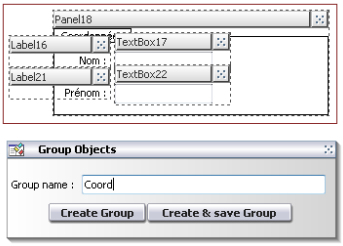
---

**Select All**
Selects all objects of the page
**Unselect**
Unselects all objects of the page.

---




**Group objects**




Creates a group of objects. This group can only be created on the page, click on "Create Group" or created and saved for future use in another page or another script.
**Ungroup objects**
Ungroup the objects in the selected group.

---

**Set Backward**
Returns the object in the foreground. Is equivalent to the Z-index property of the object (z-index 1)
**Set Forward**
Move to the first plan. Is equivalent to the Z-index property of the object (z-index-1)

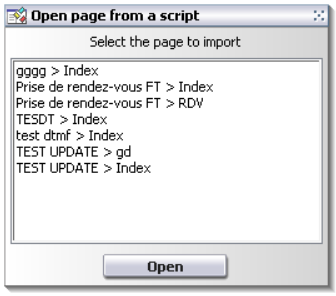
---

**Align centers**
Vertically aligns the selected objects (changes the property "Left" of objects).

**Align middles**
Aligns the selected objects horizontally (changes the property "Top" of objects).

### 3.2.4 Template Menu

**Pages …**

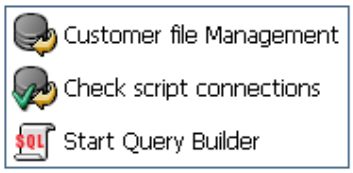

Open page from a script
Select the page to import
gggg > Index
Prise de rendez-vous FT > Index
Prise de rendez-vous FT > RDV
TESDT > Index
test dtmf > Index
TEST UPDATE > gd
TEST UPDATE > Index
Open

Can import a page from another script. Pages available in this screen are pages that have been saved with the option "Share this page, see Menu page.

**Groups …**
Can import a group that has been saved, see Edit menu.

**Delete Group …**
Allows you to delete one or more groups of objects saved. See Edit menu.

### 3.2.5 Menu Connections




**Customer file management**
To start the application fields of connections to the script database. This application is described in Annex Connections and client tables management.

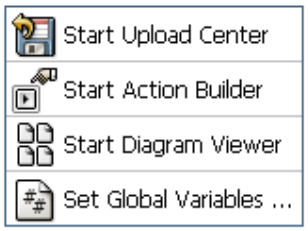
**Check script connections**
To start the application for verification of the fields connections to script database. This application is described in Annex Checking connections.

**Start Query builder**
To start the SQL management console. This application is described in Annex Query

**The three modules: Customer file / Script connections / Query builder may be executed in separate windows by using the CTRL key when clicking the menu item.**


Vocalcom - 7 rue de Tilsitt – 75017 Paris - Tél. : +33 1 55 37 30 50 – Fax : +33 1 55 37 30 51 – www.vocalcom.com
SA au capital de 5 000 000 € – RCS Paris B 401 973 631 – TVA Intracommunautaire : FR 27401973631

### 3.2.6 Menu Modules




**Start upload center**
This module allows you to download files from the web server. Files styles (css), image files (. Jpeg,. Gif,. Png), JavaScript files (. Js), and audio file (. Wav).
The downloaded files are available for all users of the company.

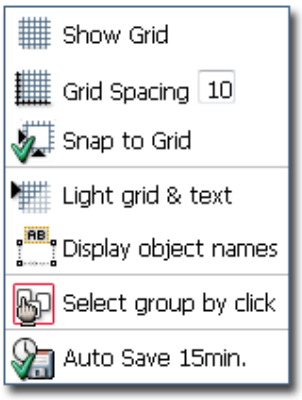
**Start Action Builder**
To start the generation of actions. The actions of the publisher will then be applied to objects of different events of the script.
This application is described in Annex Action Builder.

**Start Diagram Viewer**
To start the application diagram. This application described in Annex Diagram Viewer allows a arborescent representation of the pages of the script.

**Set Global Variables**
Opens the window creation and management of global variables of the script. See appendix "Defining global variables "

**The three modules: Upload Center / Action builder / Diagram Viewer may be executed in separate windows by using the CTRL key when clicking the menu item.**

### 3.2.7 Preferences Menu




**Show Grid**
Defines whether the grid tracing of the page should be displayed or not. The grid allows you to align objects.

**Grid spacing**
Enter the value (in pixels) of the spacing of the grid.

**Snap to Grid**
Enables or disables the automatic alignment of objects between them.

**Light grid & text**
Allows you to change the color of the grid and the names of objects. When the background is dark, it is preferable to use a clearer color grid, in this case, select this option.

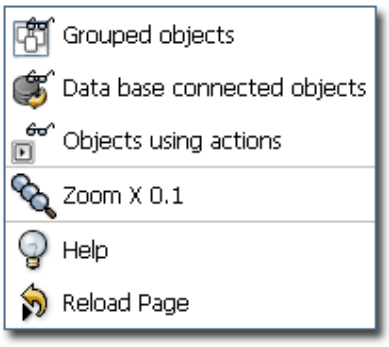
**Display object name**
Displays the name of the object

**Select group by click**
Allows you to specify if you want to select all objects in a group when you click on one of its members.

**Auto Save 15min.**
Enables automatic backup of the current page every 15 minutes.
A gauge at the bottom left of the screen allows you to know the status of the next backup.

Preferences are automatically saved when saving pages.

### 3.2.8 Tools Menu

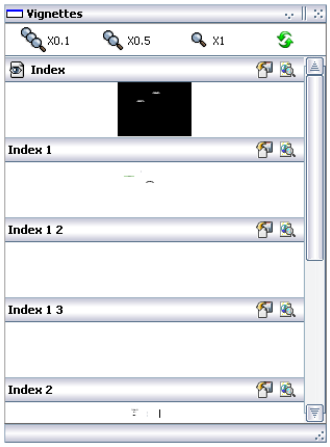



**Grouped objects**
A click on this item allows you to highlight objects grouped on the page.

**Data base connected objects**
A click on this item allows you to highlight the objects connected to a database.

**Objects using actions**
A click on this item allows you to highlight the item using an action.

**Zoom x 0.1**
Displays the screen below.




This screen allows you to preview each page of a script. The pages are displayed in 3 possible levels: x0.1, x0.5, x1.

The two icons to the right, in front of each name of the page allow you to open the page in the editor, or to preview this without being obliged to open it.

**Reload Page**
Refreshes the current screen. A backup is available before refresh.

## 3.3 The menu bar

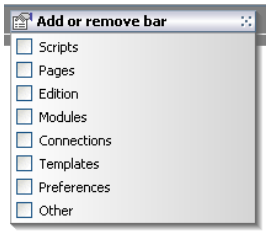
The menu bar allows you to customize the interface and presentation of menus according to your preferences. Once placed in the bar, all menu items are visible and can be customized (show / hide).

### 3.3.1 Move menus in the bar.

There are 2 ways of moving a menu into the bar :

- Click on the right corner of the selected item. The icon of the mouse displays a '+' mean that you will add a menu bar.

  Click on the bar itself. A window appears with the list of menus available. Check the boxes to add or remove its menu bar.




### 3.3.2 Removing menus in the bar.

There are two ways to remove a menu from the bar :

- Drag and drop the menu (using the area ) on the workspace, then click the pop-up.
- As in Chapter 3.3.1 above, uncheck the box corresponding to the menu to remove.

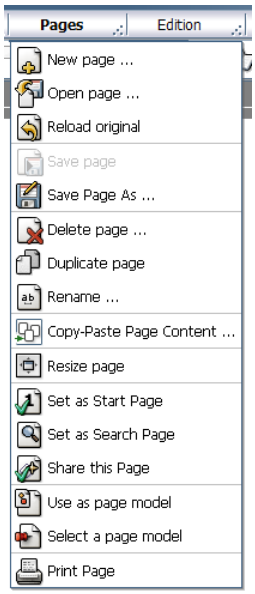
### 3.3.3 Flotting Menus.

Placed in the menu bar can be moved to turn on the workspace and be represented in the form of pop-up. These pop-up are resizable.

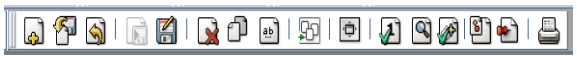
### 3.3.4 The 3 types of menus representation.

Example on the menu 'Pages'

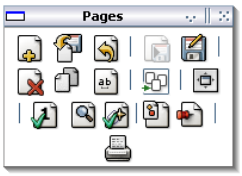
**Standard**




**bar**

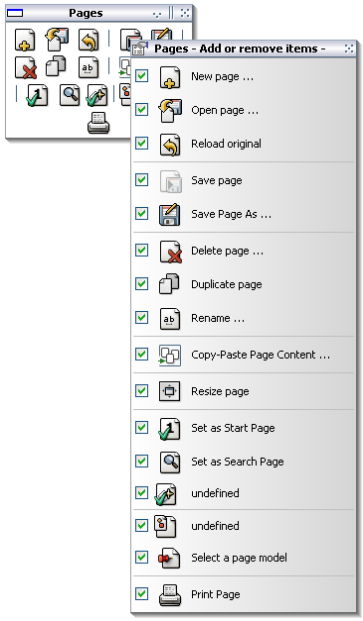


**Flotting**




### 3.3.5 Customizing the menu bar.

Once placed in the menu bar, or floating, it is possible to customize the content. To do this, right click on an item. Appears when a new window in which are displayed all menu items. Check or uncheck the items you want to keep.

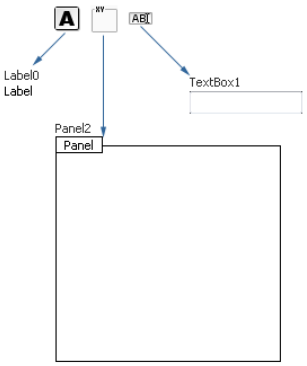


### 3.3.6 Particular case of the item menu.

The menu item cannot be deleted. It is always present in the menu bar or menu. This pop-up does not have the cross to close it.

## 3.4 The available objects

### 3.4.1 Summary




**Label**

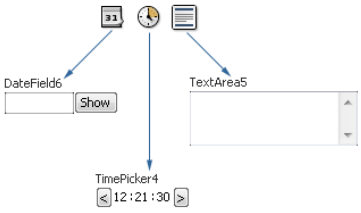
Displays text on the screen. This text can be static, evaluated, or from a column in the client file.

**Panel**

Displays a panel. This panel can be used to consolidate some functions of the page.

**Textbox**

Can enter and / or display a text on a line. This text can be static, evaluated, or from a column in the client file.




**DateField**

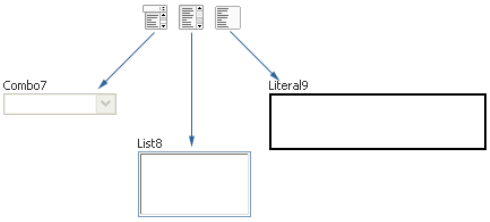
Lets you enter a date in a defined format. A button can display a calendar and select the date by clicking on it.

**TimePicker**

Allows you to enter a time in AM / PM or 24h.

**TextArea**

Can enter and / or display a complete text on several lines. This text can be static, evaluated, or from a column in the client file.




**Combo**

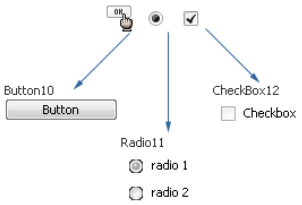
Can display a drop-down list and select a single item. The contents of this list may be static, dynamic (from the result of an SQL query).

**ListBox**

Can display a drop-down list and select one or more items. The contents of this list may be static, dynamic (from the result of an SQL query).

**Literal**

This area allows the input of HMTL code and / or JavaScript.




### Button

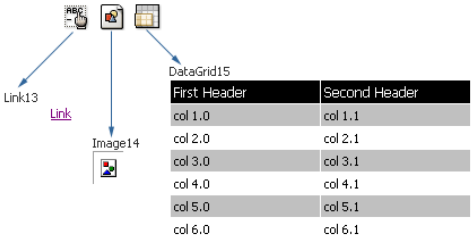
Can display a button. It is generally used to trigger an action or type of navigation or type of JavaScript.

### Radio

Make a single choice from a list of possibilities. The list of items is only static.

### CheckBox

Can make several choices from a list of possibilities. The list of items is only static.




### Link

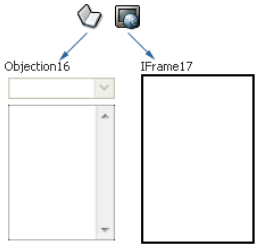
Displays a hyperlink.

### Image

Can display an image. Either a default image supplied by Vocalcom. Or a downloaded image. To download an image, use the 'Modules'>' Start Upload center. Once the images are uploaded they are available in the property 'Image' of the 'Appearance' tab.

### DataGrid

Allows you to view a table of data. The columns and data displayed are the result of the query specified in the property 'Request' tab of the 'Data'.




### Objections

Allows you to view a list of objections (one point = one). The contents of this list may be: static self, itself from the result of a query.

### Iframe

Allows you to view site in another window, called Iframe. The website address is configurable in the property 'url' of the 'Data'.

### Agenda

Allows you to view the agenda Hermes.Net. For more information about this application, please refer to the documentation 'Hermes.Net Agenda V4.

### CreditCard

Lets you enter and validate a credit card number.

### 3.4.2 Notation, and use of objects specificities

**Notations** : All objects placed on a page has a name (unique identifier on the page). This name is available through the Property tab 'Objects'>' Name '.
To access objects Hermes.Net use the notation: '@' (the rating for the global variables is:'$').

**Examples:**
If you want to change in using an action, the property 'color' of a field 'Label'. It will use the following syntax: @ IdLabel.style.color = 'red';
To change the text of a input field, use the syntax: @ IdTextBox.value = 'My new text';

## 3.5 Properties Tabs

The properties tabs provide access to properties of the page and objects contained therein. Depending on the type of object (or page), the specific tabs are displayed or hidden.

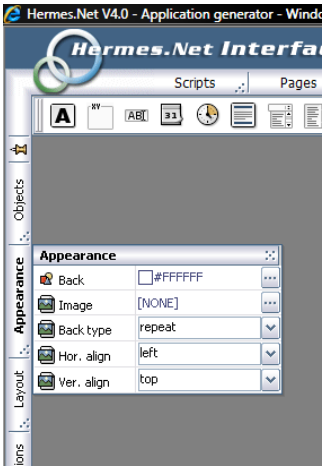
### 3.5.1 Use.

The tabs of properties can be displayed in two ways: fixed or floating. To switch from one mode to another representation, the user clicks on the icon 'pin' at the top of the tab.
In fixed mode, a fixed area to the left of the screen is reserved for placing the tab. The work area is shifted to the right of the screen to make room for content.
In floating mode, the contents of the selected tab is displayed temporarily left.

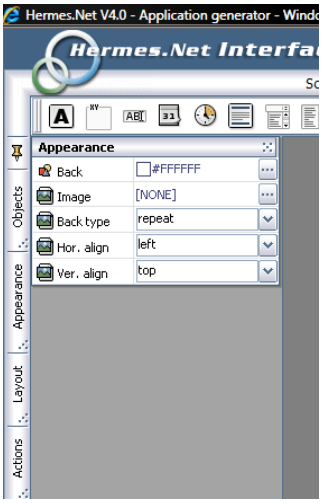
In the same way as for the menus, the content of the tabs can be placed in the form of pop-up on the workspace. To do so this click on the area of the tab.

### 3.5.2 The different modes of representation.

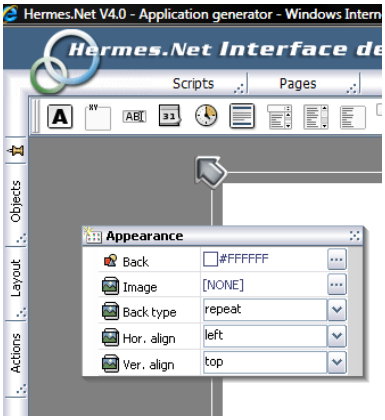
**Flotting**




**Fixed**




**Pop-Up**



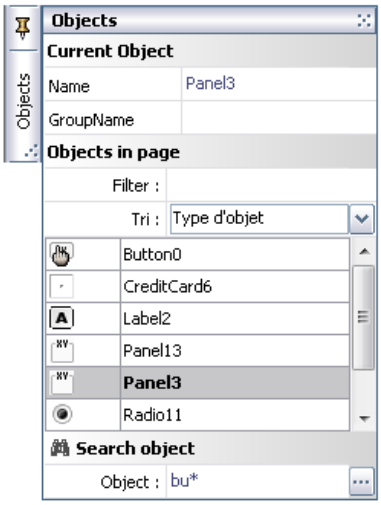
### 3.5.3 The different properties tabs.

A. **Objects**
B. **Appearance**
C. **Font**
D. **Behavior**
E. **Data**
F. **Layout**
G. **Actions**

## A - Objects

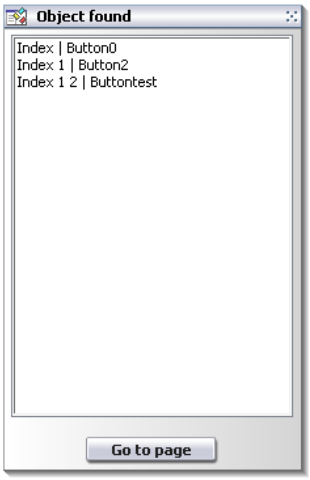
This tab is not a tab property itself. This is a tab for all items of the page.

It lets you rename objects in the page, to name groups of objects, display the list of page objects, filter and sort this list, or search an object in the script .




By clicking on an item in the list, you can put the focus on it.

The “Search” object allows you to search an object across the script. The wild card used is the asterisk (*). Once the search query is done, a pop-up containing the results is displayed on the screen. This pop-up is shown below :

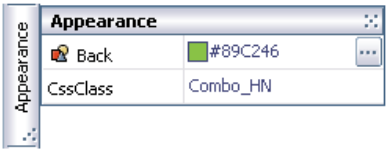



The list displayed after a search shows the name of the page where the object are and the object name.
You can load the page by clicking on the “Go To Page”.

It is possible to group objects by selecting objects with the mouse and typing directly into a group name in the 'Group Name'. This avoids going through the Edit menu> Group object.

## B - Appearance

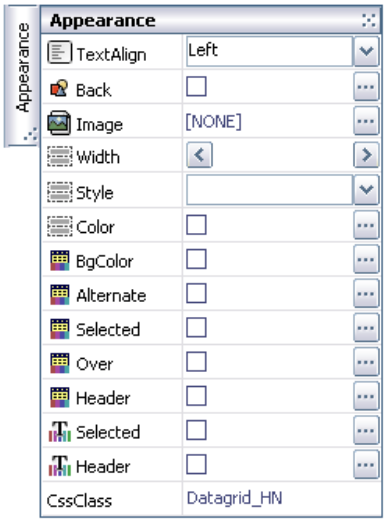
Ex : ComboBox




This tab allows you to define the presentation of the object. The possibilities for customizing objects depend on the type of object. This can range from a Combo Box object, or there are only two properties to the object datagrid which has many more properties.

For these reasons we will not describe all the properties of the 'appearance', most understandable and visual.

Ex : DataGrid

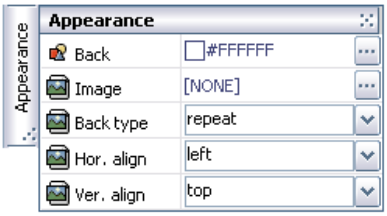



For these reasons we will not describe all the properties of the 'appearance', most understandable and visual.

List of objects with these properties:

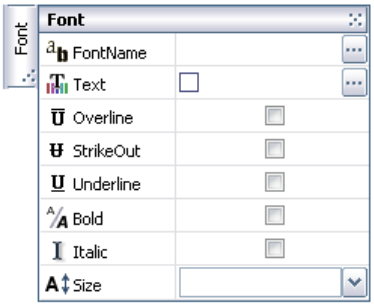
*Page, Label, Panel, TextBox, DateField, TimePicker, TextArea, ComboBox, ListBox, Literal, Button, Radio Button, CheckBox, HyperLink, Image, Data Grid, Objections, Iframe, Agenda, Credit Card.*

Ex : Page




It is possible to use the CssClass property to refer to a style file that you downloaded. See Appendix Create your own style file.

## C - Font




This tab allows you to define the presentation of the texts of the object. The possibilities for customizing objects depend on the type of object.

The Font tab is identical for all objects that use it.

List of objects with these properties:

*Page, Label, Panel ,TextBox, DateField, TimePicker, TextArea, ComboBox ,ListBox, ~~Literal~~, Button, Radio Button, CheckBox, HyperLink, ~~Image~~, Data Grid, Objections, ~~Iframe~~, ~~Agenda~~, Credit Card.*

It is possible to use the CssClass property to refer to a style file that you downloaded. See Appendix Create your own style sheet.

## D - Behavior

This tab allows you to define the behavior of objects.
We will see the properties object by object because it is very dependent on the type of action possible with the object.

List of objects with the 'Behavior' property:

*~~Page~~, Label, Panel ,TextBox, DateField, TimePicker, TextArea, ComboBox ,ListBox, Literal, Button, Radio Button, CheckBox, HyperLink, Image, Data Grid, Objections, Iframe~~, Agenda, Credit Card.*

Properties common or standard:

**Export Order** : This property has been added to meet the needs of reading through pages of a table reading Braille. Indeed the software refer to the table Braille information in the order they appear on the page. This property allows us to schedule information to be displayed on a table Braille. For use standard it can be ignored and left to its default.

**Access Key** : Allows you to assign a hotkey to the object. Example: If the property of a TextBox object is set to 'Z', when the user presses the keys' ALT '+' Z ', the focus will be on this subject.

**ToolTip** : Used to display a tooltip containing the text entered in the property when the user places his mouse on the object.

**TabIndex** : Allows you to schedule items to a seizure. The user, using the keyboard 'TAB' to move from one field to another in the order of the TabIndex property.

**Enabled** : Allows you to disable the function of the object. For example, a TextBox object with the Enabled property unchecked, cannot enter text. From the disabled list cannot choose an Item, etc..

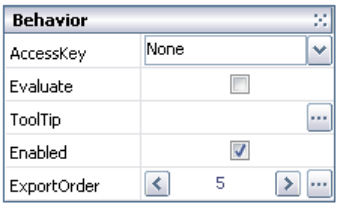
**Evaluate** : Allows you to specify if the text property of the object must be, if it contains a global variable, replaced by the contents of the variable. For example, if the text of the object is: "Hello $ Contact ', if the property 'Evaluate' is checked, then the user script has the text "Hello Mr. Vocalcom" (Yes, of Naturally, the variable contains $ Contact 'M. Vocalcom '.

**AllowNull** : When this box is not checked, entering a value will be required to validate the page

.

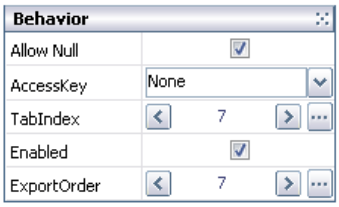
The items listed below are the views of the properties above.

**Label**

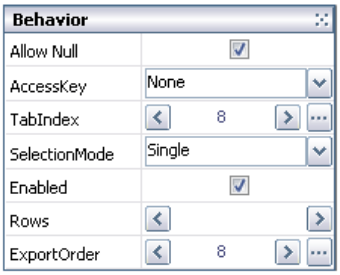



**Panel**

**ComboBox**

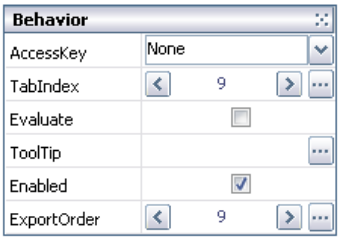



**ListBox**

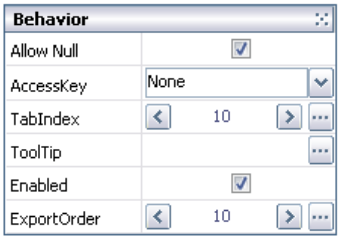



**Literal**

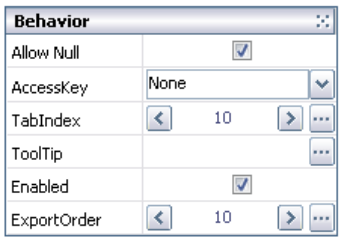
**Button**



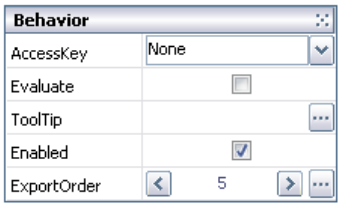

**RadioButton**



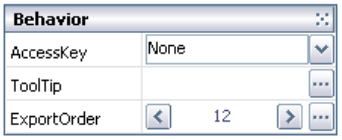

**CheckBox**



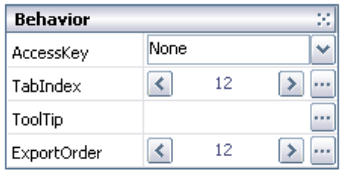

**Link**



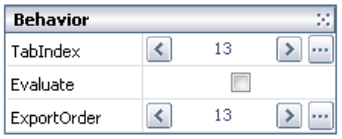

**Image**



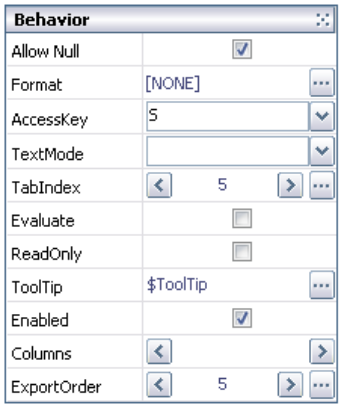

**Objections**



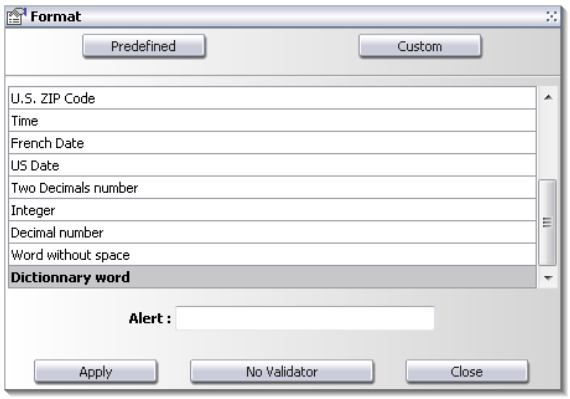

**Iframe**



### TextBox




**Format** : You can define a format for entering the field. By clicking on the icon the following screen appears.




This screen allows you to select an existing format, or you create your own format using the button 'Custom'.
The format for the validation of an expression is generated by a 'regexp' javascript. For those wishing to create their own format, it is interesting to visit:
http://www.regular-expressions.info/javascriptexample.html

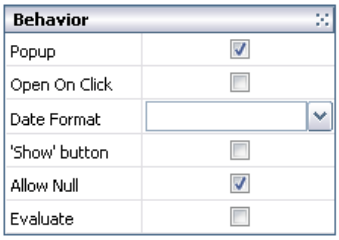
When the defined format is not respected, the system displays the warning contained in the 'Alert' .

**Text Mode** : Allows you to select the attachment type 'Password'. (text illegible).

**ReadOnly** : The area is read-only.

**Columns** : Allows you to adjust the width of the input area.

### DateField



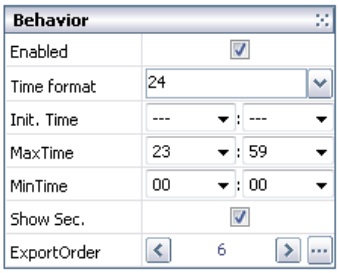

**Popup** : Allows you to open the calendar in a new browser window (Pop-up). Unchecked, the calendar appears in the same page, under the input area.

**Date Format** : Used to specify in what format should be the date shown. A list of formats is available but the input box allows you to enter your own format.
d : Numeric day.
dd : 2 digits numeric day
M : Numeric month.
MM : 2 digits numeric month
MMM : Text month.
y : Year in x digits
yy : Year in 2 digits
yyyy : Year in 4 digits

**Open On Click** : Allows you to open the calendar when the user clicks on the box. Otherwise the calendar is visible by clicking the 'Show'.

**'Show' button** : To show or hide the button of the selection date.

### TimePicker




**Time format** : to select the time display 24h ou 12h.

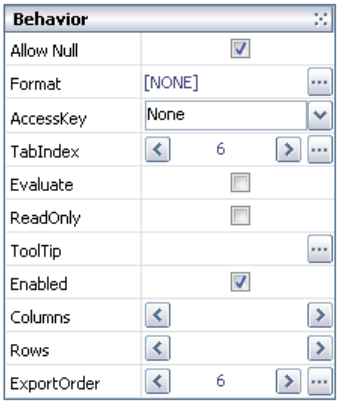
**Init time** : Predefines an hour.

**Max. Time** : Can limit the display to max value.

**Min. Time** : Can limit the display to a min value.

**Show Sec.** : Shows or hides the seconds.

### TextArea



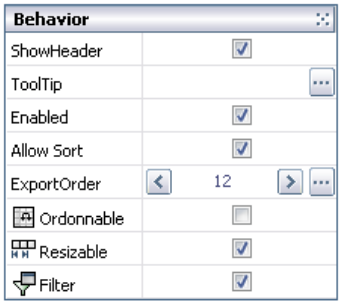

**Format** : Idem Objet TextBox.

**ReadOnly** : The area is read-only.

**Columns** : Allows you to adjust the width of the input area.

**Rows** : Allows you to adjust the height of the input area.

### DataGrid



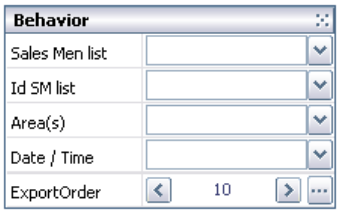

**Allow Sort** : Displays icons sorting columns.

**Resizable** : Allow resizing of table columns.

**Filter** : Allows the display of field 'filter' at the top of columns.


Vocalcom - 7 rue de Tilsitt – 75017 Paris - Tél. : +33 1 55 37 30 50 – Fax : +33 1 55 37 30 51 – www.vocalcom.com
SA au capital de 5 000 000 € – RCS Paris B 401 973 631 – TVA Intracommunautaire : FR 27401973631

### Calendar

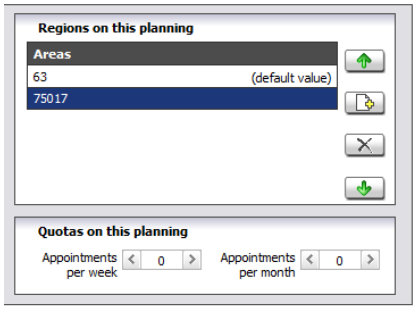



**Sales Men list** : Allows you to predefine commercials at the opening of the agenda. The separator is'; '.

- It is possible to use a global variable for this property. Indeed, the commercial to be displayed is often a function of the script. It is a simple script action to update the variable containing the list of visible commercials on the agenda.

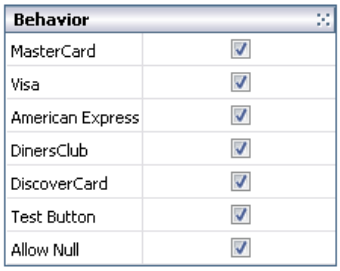
**Id SM(Sales Men) list** : same as here above, but we're using the commercial ID and not their names.

**Area(s)** : Allows you to specify one or geographic areas to filter commercial. These values refer to the screen region in the commercial administration Hermes.Net




**Date / Time** : Allows you to specify a date and time for the launch of the calendar. The format used is as follows: yyyymmddhhmm.

### CreditCard




**MasterCard > DiscoverCard** : Allows you to specify which type of credit card may be used.

**Test button** : Show or not the button to validate the input of information on the card number.

## E - Data

This tab allows you to connect the objects in the fields of different databases and data tables.
Regarding the database fields, it is possible to use the applications and client management tables defined in the Annex.

On some objects, properties about the display and save have been added in this tab. Ex: Associate Var. which can affect or save the content of the object in a global variable rather than a table.

**Liste des objets disposants de l'onglet 'Data' :**

*~~Page~~, Label, ~~Panel~~ ,TextBox, DateField, TimePicker, TextArea, ComboBox ,ListBox, ~~Literal~~, ~~Button~~, Radio Button, CheckBox, ~~HyperLink, Image~~, Data Grid, Objections, ~~Iframe~~, ~~Agenda~~, Credit Card.*

Version 4 of Hermes. Net allows you to select a column of the table to display customer data (RD) and another for data backup (WR). This can be handy when you do not want to change the original customer table. The properties with the suffix 'RD' concerns the display, and the suffix 'WR' the backup.

**Properties common or standard:**

**Cx RD** : Defines the connection string (created in the administration) to be used for reading the field.
**Table RD** : Set the table where the column shoulb be displayed.
**Txt Col. RD** : Display column.
**Val Col. RD :** Some items have a text (displayed in the object) and value (not displayed, but indexed text). These two properties of the object can be treated separately.
These objects are: Combo, Listbox, RadioBox, CheckBox, Objections.

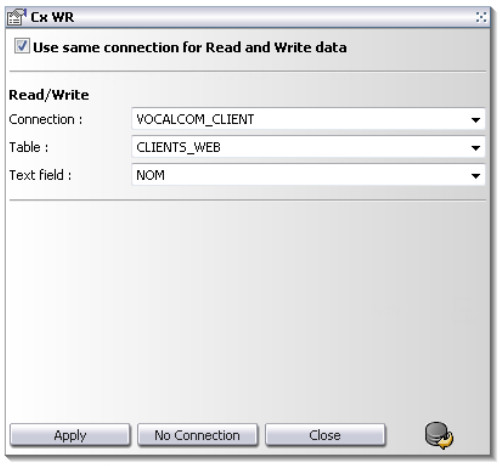
**Cx WR** : Defines the connection string (created in the administration) to be used to save the field.
**Table WR** : Set the table where the column shoulb be displayed.
**Txt Col. WR** : Column used to save.
**Val Col. WR :** Some items have a text (displayed in the object) and value (not displayed, but indexed text). These two properties of the object can be treated separately.
These objects are: Combo, Listbox, RadioBox, CheckBox, Objections.

**Associate Var.** : This property allows you to assign the value read from the table a global variable of your script.


Vocalcom - 7 rue de Tilsitt – 75017 Paris - Tél. : +33 1 55 37 30 50 – Fax : +33 1 55 37 30 51 – www.vocalcom.com
SA au capital de 5 000 000 € – RCS Paris B 401 973 631 – TVA Intracommunautaire : FR 27401973631

**Pop-up with connected objects :**

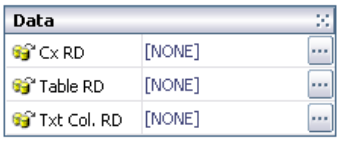
The connection properties of objects are configurable using following pop-up:



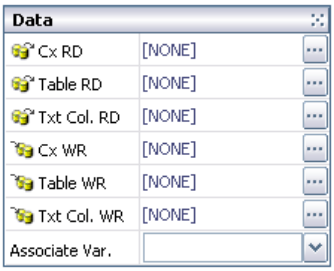

This window lets you specify which fields to use, read / write for the connection.
Warning: If you do not want to connect the object reading and writing it is necessary to remove the box: 'Use the same connectio for read and write data, and then specify two different connections or disable a connection (reading or writing).

The items listed below only show the properties shown above.

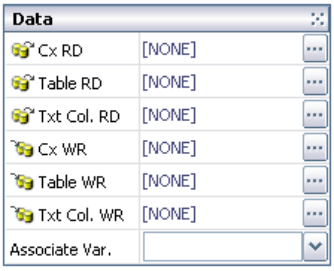
**Label**



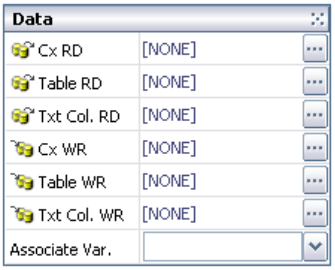

**TextBox**




**DateField**

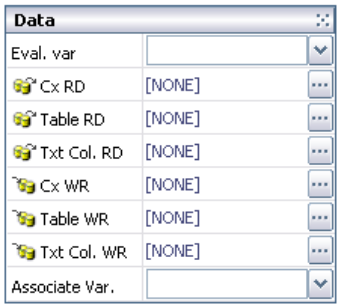



**TextArea**



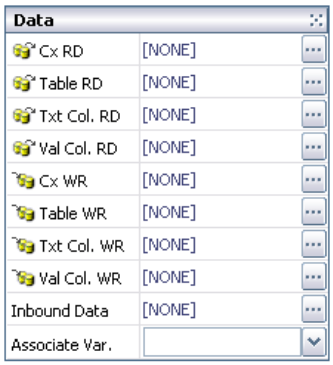

**Credit Card**

### Time Picker



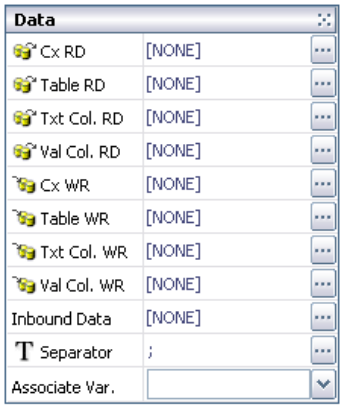

**Eval. Var** : allows you to set the time of the object to the value contained in a global variable.

### Combo




**Inbound Data** : This property is used to define the content of the Combo. The configuration window is documented lower.

### ListBox / RadioBox / CheckBox

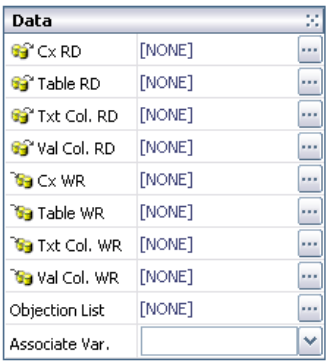



**Inbound Data** : This property is used to define the content of the Combo. The configuration window is documented lower.

**Separator** : When property 'Multi-Select' is active, it is necessary to specify the separator used to separate the values written into the table. Same if the value is read from the column of the table.

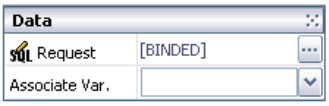

Vocalcom - 7 rue de Tilsitt – 75017 Paris - Tél. : +33 1 55 37 30 50 – Fax : +33 1 55 37 30 51 – www.vocalcom.com
SA au capital de 5 000 000 € – RCS Paris B 401 973 631 – TVA Intracommunautaire : FR 27401973631

### Objections




**Objections list** : This property is used to define the content of the objections. The configuration window is documented lower.

### DataGrid



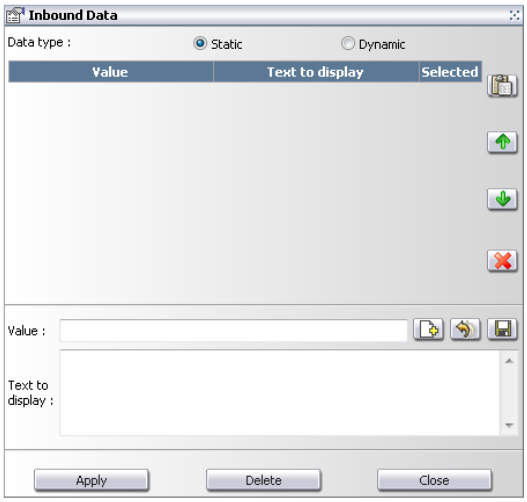

**Request** : This property specifies the contents of the table. This content is displayed according to a query on a table. The columns displayed depend on the fields returned by the query.

- The request may be generated via the query module editor described in Appendix.
- Request property may contain a global variable of the script. This use is very convenient for the use of dynamic queries.

**Window property 'Inbound Data'**

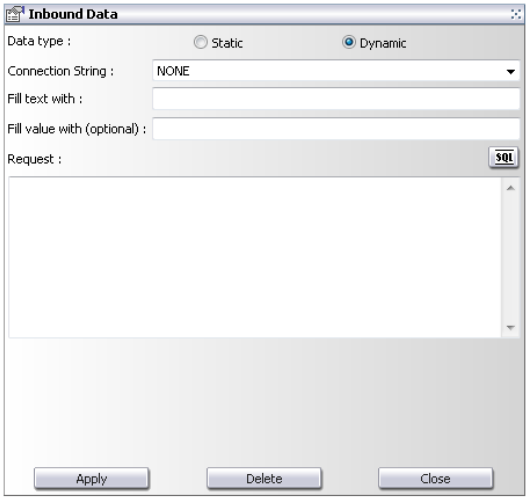
There are 2 ways to complete the content of the objects' Combo ',' ListBox ',' Radio Box ',' Check Box '. The first is to provide a static list of values. Static values each being composed of a (usually digital) and a text which is displayed in the object.
To configure a static content, check the 'Static' in the settings window.
The use of a value plus a text for the items used to store a numerical value and not text. This may be more convenient for the reporting and processing of data in back office.

**Static content**




This window allows you to enter items to be displayed.
As explained above, the field value is optional and takes the value of text field when it is omitted.
The items in the list may be ordered using the green arrows' high '' low '
.

**Dynamic content**




This screen allows you to create a dynamic list in two columns in a table.

Note on the automatic positioning of an item based on the content of a column. Hermes.Net uses the contents of the property 'Value' of object to be positioned automatically on an item. When property 'Val Col.RD is left empty, the column' Col Txt. RD 'is used.

## F - Layout

This tab allows you to define display properties and presentation objects.

List of objects with the 'Data' :

*Page, Label, Panel ,TextBox, DateField, TimePicker, TextArea, ComboBox ,ListBox, Literal, Button, Radio Button, CheckBox, HyperLink, Image, Data Grid, Objections, Iframe, Agenda, Credit Card.*

Properties common or standard:

**Width** : Largeur de l'objet.
**Height** : Hauteur de l'objet.
**Top** : Position Y de l'objet dans la page.
**Left :** Position X de l'objet dans la page.
**Zindex** : Position Z de l'objet dans la page.
**Visible** : Affiche ou cache l'objet.

Specific properties by object :

### Page

**Left Margin** : Sets the left margin of the page.
**Top Margin** : Defines the top margin of the page.

### TextArea

**Wrap** : allows aotomatic wrapup.

### DataGrid

**Header** : Sets the height of the header table.
**Line** : Defines the height of the table rows.
**CellSpacing** : Sets spacing between table cells.
**CellPadding** : Sets the internal margin of the table cells.

### Iframe

**Margin Width** : Sets the internal margin left and right frame.
**Margin Height** : Defines the internal high margin and low.


Vocalcom - 7 rue de Tilsitt – 75017 Paris - Tél. : +33 1 55 37 30 50 – Fax : +33 1 55 37 30 51 – www.vocalcom.com
SA au capital de 5 000 000 € – RCS Paris B 401 973 631 – TVA Intracommunautaire : FR 27401973631

## G - Actions

This tab allows you to define the actions performed on the of the object event and / or define the jump of pages to perform.

List of objects with the 'Data':

*Page, ~~Label~~, ~~Panel~~ ,TextBox, DateField, ~~TimePicker~~, TextArea, ComboBox ,ListBox, Literal, Button, Radio Button, CheckBox, HyperLink, Image, Data Grid, ~~Objections~~, Iframe, Agenda, Credit Card.*

Explanations of different possible events on the objects:

**OnLoad** : Draws the specified action for loading the object.

**OnClick** : Launches action on click.
**OnDblClick** : Launches action on double click.

**OnMouseOut** : Launches the action when the mouse cursor leaves the object.
**OnMouseOver** : Launches the action when the mouse cursor goes over the object.

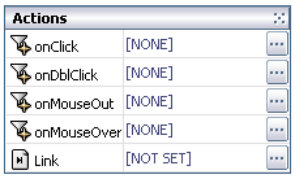
**OnChange** : Launches action when value changes.
**OnSelect :** Launches the action when one of the items of the object is selected.
**OnKeypress** : Launches the action when pressing a keyboard button.

Note that there is a JavaScript method to add other events to an object. For example, to add the event 'onMouseOver' TextBox to an object named 'MyTextBox', we add an action in the following code: @ MyTextBox.attachEvent ( 'onMouseOver', MyFunction)

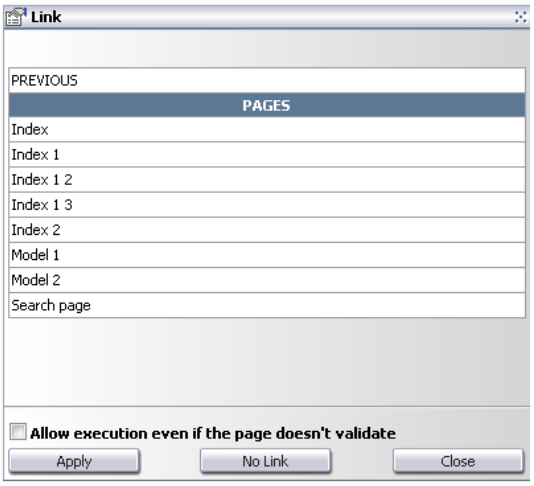
Specific properties by object :

---

### Button




**Link** : Allows you to load a page of script when clicking on the button.

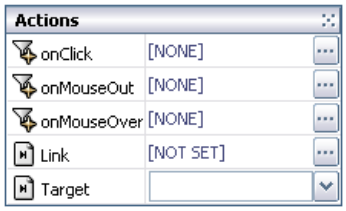



**PREVIOUS** page allows you to return to the previously loaded page.

Select the checkbox at the bottom of the window if you want to navigate to the destination even if the validation of the current page is incorrect.

- The shares put on the events of the object are executed before the start of the page defined in the property 'Link'.

### Link / Image




**Link** : see above.

**Target** : Allows you to specify what window should be launched page.

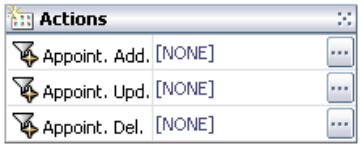
_Blank : New beowser window.
_Parent : Opening script page.
_ Search : same as _Blank on IE7 and +
_Self : Curent window
_Top : Main window.

- The two modes _Parent and _Top are not to be used with WorkSpace Agent.

### Iframe

**Url** : Address of the page to load in the iframe. May contain a global variable.

### Agenda




**Appoint. Add** : This event is called when the user adds an appointment in the calendar.

**Appoint. Upd** : This event is called when the user modifies an appointment in the calendar.

**Appoint. Del** : This event is called when the user deletes an appointment in the calendar.

**Important notes about the events on the calendar.**
At an event, the appointments are stored directly on the subject "of date". It is possible to access the members of the object "an appointment", but it should be noted that the information stored there correspond to the information of the appointment of the last event lifted.

Most of the information of appointments that were available in Hermes.Net V3 were retained for compatibility reasons. However, due to changes in architecture Hermes.Net V4, certain changes have been made.

We list below the information available on the subject "Making appointments and to facilitate work during migration :

black: information retained,
green: information added,
In blue: the amended data
in red: the deleted information.

In the following sections, we consider the subject "an appointment" named "@ Calendrier0 in interface designer.

**Information about appointments**

@ Calendrier0.appointmentId: id of the appointment
@ Calendrier0.personId: id trade for which an appointment has been made
@ Calendrier0.contactId: Guid of the contact in the Address Book
@ Calendrier0.calendarId: id schedule defined in the Admin
@ Calendrier0.typeId: Id-type appointments as defined in the Admin
@ Calendrier0.appointmentSubject: about the appointment
@ Calendrier0.appointmentDescription: description of the appointment
@ Calendrier0.appointmentBeginDate: start date of appointment (Hermes format)
@ Calendrier0.appointmentEndDate: end date of appointment (Hermes format)
@ Calendrier0.appointmentStartTime: starting time of the appointment (Hermes format)
@ Calendrier0.appointmentEndTime: end time of the appointment (Hermes format)
@ Calendrier0.appointmentDuration: duration of appointment (in milliseconds)
@ Calendrier0.appointmentWeekNumber: number of weeks of the appointment
@ Calendrier0.appointmentNotes: notes entered for the appointment
@ Calendrier0.creationDate: date of the appointment
@ Calendrier0.userId: oid of the agent that generated the event
@ Calendrier0.isDeleted (true / false): indicates whether the appointment has been deleted
@ Calendrier0.isConfirmed (true): this concept no longer exists v4 value always true
@ Calendrier0.isBlocked (false): this concept no longer exists v4 value always false
@ Calendrier0.lastModificationDate: date of last modification of the appointment
@ Calendrier0.appointmentAdressType: description of the type of address of the appointment in the list of addresses defined in the admin
@ Calendrier0.appointmentAdress1: Line 1 of address of the rendezvous
@ Calendrier0.appointmentAdress2: Line 2 of address of the rendezvous
@ Calendrier0.appointmentZipCode postal code of the address of the rendezvous
@ Calendrier0.appointmentCity: city of the address of the rendezvous
@ Calendrier0.appointmentCountry: country of the address of the rendezvous

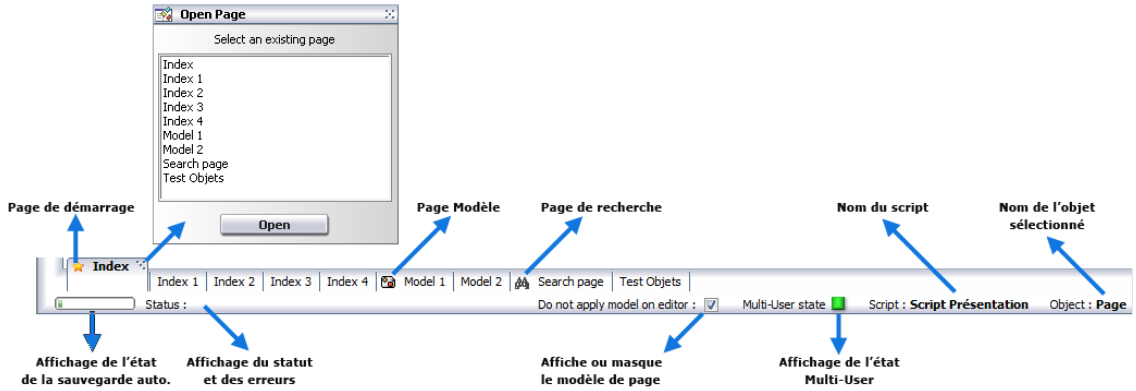
**Contact onformation**

Contact information that was available in V3 have been deleted.
It is now possible to find all this information (and more) through information "@Calendrier0.contactId " and the editor command action "Contact Info" described below in Chapter 9.5.9.
Contact via the object recovered, it is possible to have access to all the old information, since a contact can now have multiple addresses, phone number and several emails.
The information is removed :

@Calendrier0.contactName
@Calendrier0.contactFirstname
@Calendrier0.contactCompanyName
@Calendrier0.contactFunction
@Calendrier0.contactMobilePhoneNumber
@Calendrier0.contactMainPhoneNumber
@Calendrier0.contactAssistantPhoneNumber
@Calendrier0.contactPersonnalEmail
@Calendrier0.contactOfficeEmail
@Calendrier0.contactAssistantEmail
@Calendrier0.contactFaxPhoneNumber

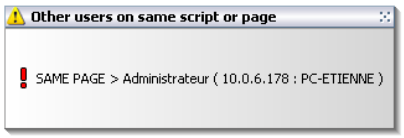
**Commercial information**
Calendrier0.personId: id commercial
@ Calendrier0.personFirstName name of the commercial
@ Calendrier0.personLastName: name of the commercial
@ Calendrier0.personAddress: commercial address (may include multiple lines) $
@ Calendrier0.personZipCode postal code of the address of the commercial
@ Calendrier0.personCity: city of the address of the commercial
@ Calendrier0.personMainPhoneNumber: main telephone number of commercial
@ Calendrier0.personMobilePhoneNumber: mobile phone commercial
@ Calendrier0.personEmail: email address of business
@Calendrier0.personFunction
@Calendrier0.personCompanyId
@Calendrier0.personWebLogin
@Calendrier0.personWebPassword
@Calendrier0.personWebConsultAllowed
@Calendrier0.personLastModificationDate
@Calendrier0.personCreationDate
@Calendrier0.personUserId

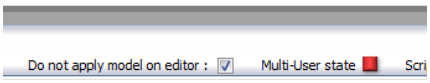
## 3.6 Status bar



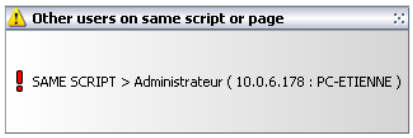

LED Multi-User State 'lets you know if other users are working

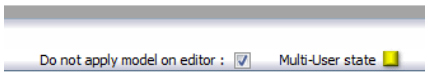
- either on the same page (red)




- Either on the same script (orange)



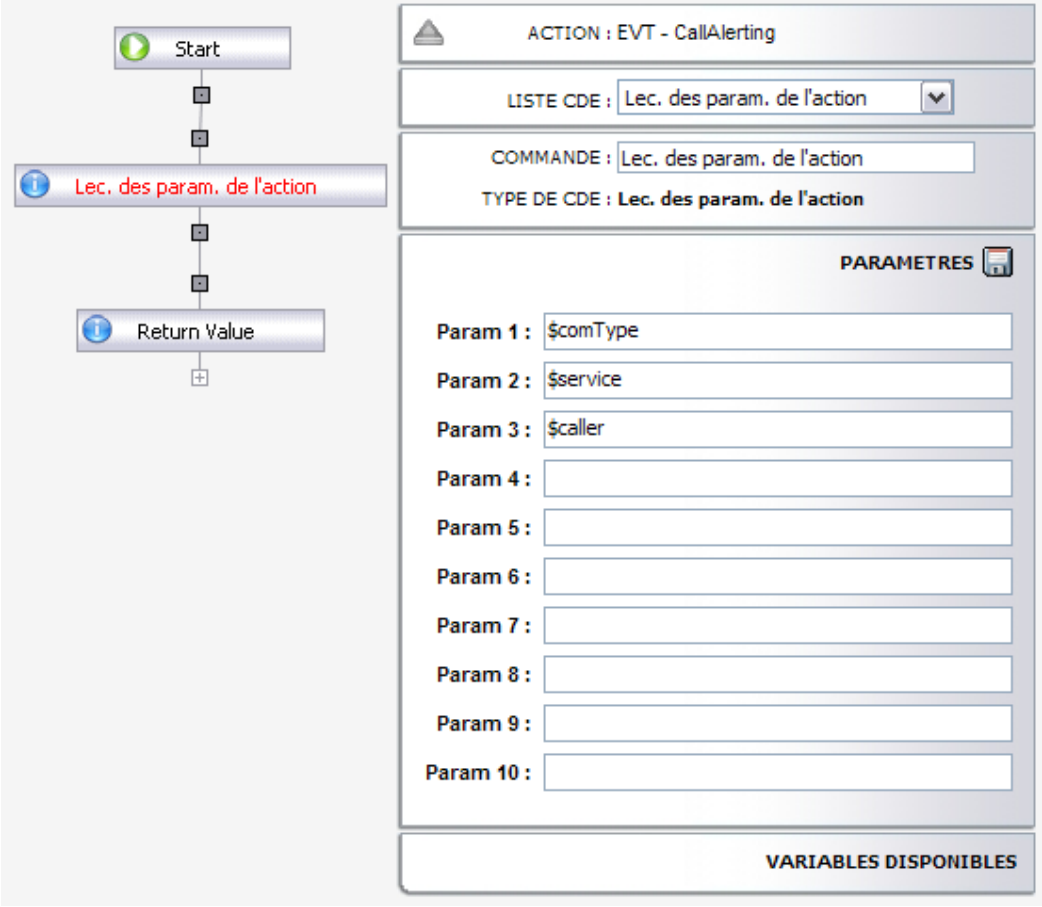

It is important to check the status of this indicator when you work on a particular page or other user is located. Indeed, if the remote user uses the automatic backup, and he leaves his post, your changes will be overwritten every 15 minutes

# 4. « CTI Events» Appendix

## 4.1 CTI event principal

At each CTI event, the workspace calls on each scripts dynamic / static / open / of the corresponding CTI event, if an action has been attached. To attach an action to an event, open the popup event management CT I: Scripts menu -> Action events on CTI.

The events can be called with parameters (for the list of parameters per event, see next section). It is possible to recover the value of these parameters in global variables using the command "read. param. of action ":

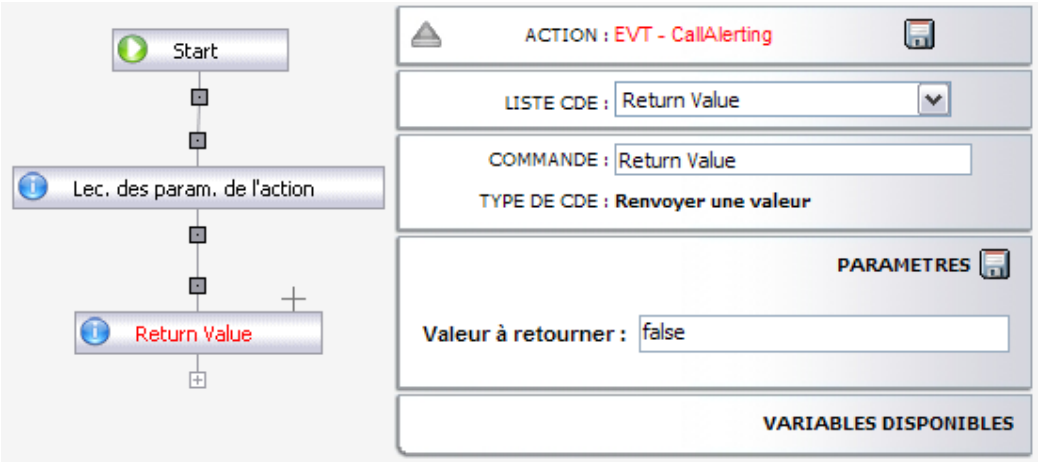



In this example we retrieve the first parameter passed to the action in the global variable "$ comType", the second in the global variable "$ service" and the third in the global variable "$ caller.

Moreover, for some events it is possible to cancel the action normally done for an event (for the detailed list of actions and their effect voidable, see next section).

This is done through the "Return value". To cancel the event should return the boolean value "false" (see screenshot below)

.




If several frames of scripts in the workspace have defined a single event, regardless of return, the action of the event all scripts will be called. As against only one script only returns false if the event is canceled.

## 4.2 List of events sent to the script by the workspace

**NOTE: are detailed here only the events corresponding to the telephone. For events scripts mails, please refer to the technical note "Interaction between the mail and the script" and the events scripts chat, please refer to the technical note "Interaction between the cat and the script".**

### 4.2.1 OnBeforeOpenStatus

Sent before opening the panel status.

- Input Parameter: none
- Event blocker: yes - prevents the opening panel status
- Called on Dynamic Pages / Static Pages / Hidden Pages

### 4.2.2 OnStatus

Sent when the agent validates the call status.

- Input parameters:
  1. Status Code
  2. Status under Code
  3. Recall date format YYYYMMDD (empty string if no recall)
  4. Telephone for the reminder (empty string if no recall)
  5. Duration of validity of the reminder in seconds (empty string if no recall)
  6. Comments submitted by the agent at the point
- Event blocker: yes - prevents the sending of the call status to ProxyCTI
- Called on Dynamic Pages / Static Pages / Hidden Pages

### 4.2.3 CallSuccess

Sent at the beginning of a call.
• Input parameters:
1. Call type. Corresponds to the property "CallInformation.ComType / 10" of AgentLink
2. Name of the campaign for which the agent receives the call
3. Telephone number of the calling / called party
• Blocking Event: no
• Called on: Static Pages / Hidden Pages

### 4.2.4 CallPreview

Sent when a record is being previewed.

- Input parameters:
    1. Campaign name
    2. Phone number linked to the record
- Blocking Event: non
- Called on: static Pages / hidden pages

### 4.2.5 CallFailure

Sent when a call fails.

- Input parameters :
    1. Type of call. Corresponds to the property "CallInformation.ComType / 10" of AgentLink
    2. Name of campaign
    3. Failed phone number
    4. Isdn error code
- Blokking event : no
- Called on : static Pages / hidden pages

### 4.2.6 ConsultFailure

Sent when consult failed.

- Input parameters :
    1. Error description
    2. Isdn error code
- Blokking event : no
- Called on : Dynamic pages / static Pages / hidden pages

### 4.2.7 OnBeforeConsult

Sent when sclicking on « consult ».

- Input parameters: none
- Blokking event : yes **– prevents the opening of consultation popup**
- Called on : Dynamic pages / static Pages / hidden pages

### 4.2.8 OnConsult

Sent when validating the consult.

- Input parameters:
    1. Consult destination
    2. Memo associated to consult
    3. Data associated to consult

- Blokking event : yes – **prevents consult from being sent out**
- Called on : Dynamic pages / static Pages / hidden pages

### 4.2.9 QueueStateChange

Sent when a change in value of the agent queue.

- Input parameters :
    1. Amount of calls in personal queue
    2. Amount of calls in principal queue
    3. Amount of calls in secondary queue
- Blokking events : no
- Called on : Dynamic pages / static Pages / hidden pages

### 4.2.10 CallAlerting

Envoyé lorsque le téléphone du TA ou celui du prospect sonne.

- Paramètres en entrée :
    1. Type d'appel. Correspond à la propriété « CallInformation.ComType / 10 » de AgentLink
    2. Campagne associée à l'appel
    3. Caller : numéro de téléphone appelé
- Evènement bloquant : non
- Appelé sur : Pages statiques / Pages cachées

### 4.2.11 CallCleared

Envoyé lors de la fin de communication.

- Paramètre en entrée : aucun
- Evènement bloquant : non
- Appelé sur : Pages statiques / Pages cachées

### 4.2.12 CallTransfer

Envoyé à  l'agent une fois le transfert abouti.

- Paramètre en entrée : aucun
- Evènement bloquant : non
- Appelé sur : Pages statiques / Pages cachées

### 4.2.13 OnBeforeTransfert

Envoyé lorsque l'agent clique sur le bouton « transfert ».

- Paramètre en entrée : aucun
- Evènement bloquant : oui – **empêche l'ouverture du panel de transfert**
- Appelé sur : Pages dynamiques / Pages statiques / Pages cachées

### 4.2.14 OnTransfert

Envoyé lorsque l'agent valide le panel de transfert.

- Paramètres en entrée :
    1. Destination du transfert
    2. Memo associé à l'appel
    3. Données associées à l'appel
- Evènement bloquant : oui – **empêche le transfert d'être effectué**
- Appelé sur : Pages dynamiques / Pages statiques / Pages cachées

#### 4.2.15 OnInfo

Envoyé lors de la réception par l'agent d'un message d'information provenant de l'acd.

- Paramètres en entrée :
  1. Niveau de gravité du message (0 : info / 1 : warning / 2 : error)
  2. Chaîne de caractère du message
- Evènement bloquant : non
- Appelé sur : Pages dynamiques / Pages statiques / Pages cachées

#### 4.2.16 OnBeforePause

Envoyé lorsque l'agent clique sur le bouton pause.

- Paramètre en entrée : aucun
- Evènement bloquant : oui – **empêche l'ouverture du popup de pause**
- Appelé sur : Pages dynamiques / Pages statiques / Pages cachées

#### 4.2.17 OnPause

Envoyé lorsque l'agent valide le type de pause.

- Paramètre en entrée : code de pause
- Evènement bloquant : oui – **empêche la fermeture du panel de pause et l'envoi du code de pause à l'ACD**
- Appelé sur : Pages dynamiques / Pages statiques / Pages cachées

#### 4.2.18 OnPauseEnd

Envoyé lorsque l'agent sort de l'état « pause ».

- Paramètre en entrée : aucun
- Evènement bloquant : non
- Appelé sur : Pages statiques / Pages cachées

#### 4.2.19 SearchModeStateChange

Envoyé lors du changement d'état du mode recherche.

- Paramètre en entrée : booléen qui vaut « true » si recherche active, « false » sinon
- Evènement bloquant : non
- Appelé sur : Pages statiques / Pages cachées

#### 4.2.20 OnRecordStart

Envoyé lorsque l'agent débute un enregistrement.

- Paramètre en entrée : mode d'enregistrement (0 : tous / 1 : agent seul / 2 : client seul)
- Evènement bloquant : oui – **annule le début de l'enregistrement**
- Appelé sur : Pages dynamiques / Pages statiques / Pages cachées

#### 4.2.21 OnEndRecord

Envoyé lorsque l'agent stoppe l'enregistrement en cours.

- Paramètre en entrée : aucun
- Evènement bloquant : oui – **annule l'arrêt de l'enregistrement**
- Appelé sur : Pages dynamiques / Pages statiques / Pages cachées

#### 4.2.22 OnHold

Envoyé lorsque l'agent clique sur le bouton de mise en attente.

- Paramètre en entrée : aucun
- Evènement bloquant : oui – **annule la mise en attente du client**
- Appelé sur : Pages dynamiques / Pages statiques / Pages cachées

### 4.2.23 OnHoldEnd

Envoyé lorsque l'agent clique sur le bouton de reprise.

- Paramètre en entrée : aucun
- Evènement bloquant : oui – **annule la fin de mise en attente du client**
- Appelé sur : Pages dynamiques / Pages statiques / Pages cachées

### 4.2.24 OnReceiveContactInfos

Envoyé lors de la réception des informations client du carnet d'adresses défini sur la campagne.

- Paramètre en entrée : guid du contact identifié dans le carnet d'adresses
- Evènement bloquant : non
- Appelé sur : Pages dynamiques

# 5. Appendix « Creating global variables »

## 5.1 Basic principle

The concept of global variables dan interface designer is very important. Indeed it is these variables that will develop advanced scripts.
A global variable is a variable, defined and initialized in the interface designer and used in the script.

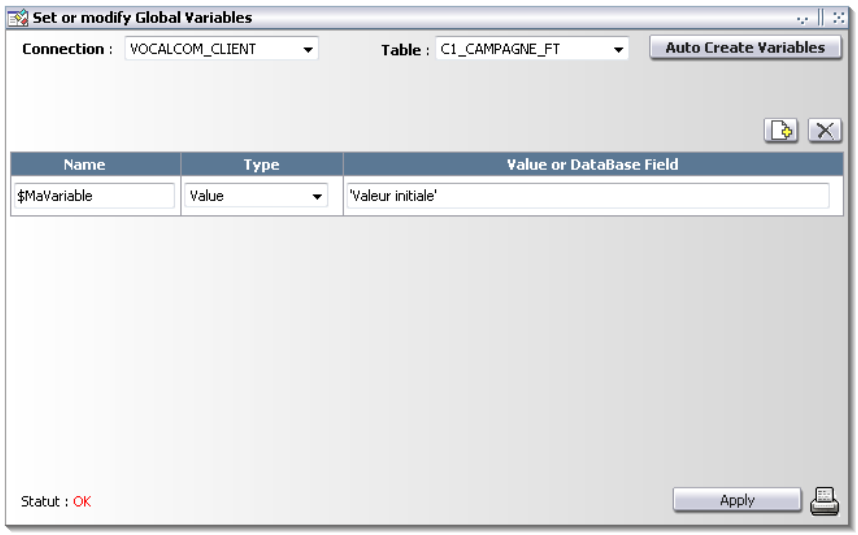
The number of global variable in a script is not limited.

The notation of variables is the '$'. All global variables must have the prefix '$'.

There are 3 ways to initialize variables in a script.

## 5.2 Free variables

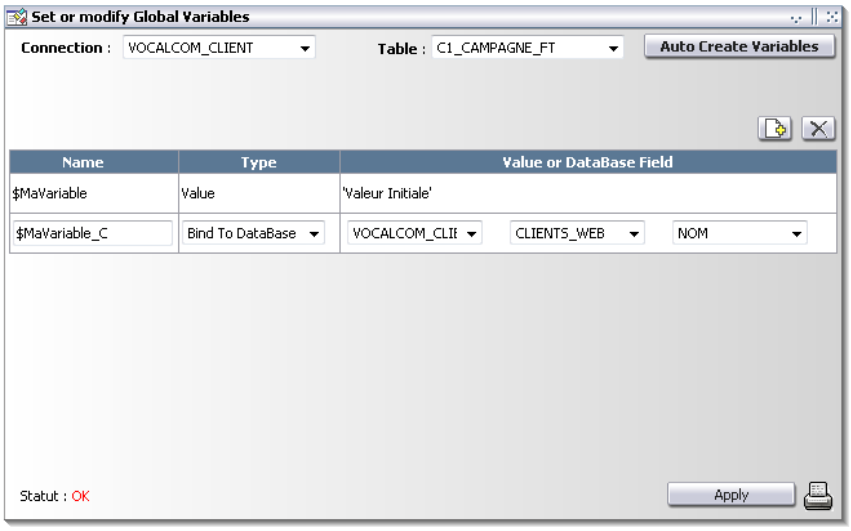
These variables are created in module 'Set Global Variables' menu available in the' Modules'.
In this mode, the user defines a name and a default value for the variable.




In the example above, we created a variable called $ myvar and this variable will value 'Value'.

## 5.3 Connected variables

These variables are created in module 'Set Global Variables' menu available in the' Modules'.
In this mode, the user defines a name and a database field. The initial value of the variable is the field content of the specified table.




In the example above, we created a variable called $ MaVariable_C and this variable will be the initial content of the column 'Name' of table 'CLIENT_WEB' for index = xxxx '(The index used is the index of the call that made up the script).

It is important to note that the table selected to initialize the global variables must contain at least one column 'INDEX'. Indeed, the system run the following query on the table.
'Select name from where CLIENTS_WEB INDEX = xxxx'. The index used is the index of the call that made up the script.

## 5.4 Dynamic Variables

These variables are created on the fly at the start of the script. The move is directly in the url of the script.
The default variables used by the script are as follows:

| | |
|---|---|
| INDICE | : Indice de la fiche en cours. |
| DNIS | : Numéro de l'appelant. |
| ANI | : Numéro appelé. |
| USER | : Code de l'agent. |
| NAME | : Nom de l'agent. |
| STATION | : Station téléphonique de l'agent. |
| ASSOCIATE | : Valeur supplémentaires recues d'un SVI. |
| MEMO | : Valeur supplémentaires passes par l'agent. |
| CAMPAIGN | : Nom de la champagne sur laquelle a été recue l'appel. |

Their values are passed to the script via the URL and launch information from the call agent and the bandeau.

To create a dynamic variable, just add it to the end of the url. Example to create a variable $ MaVariable_D with the default value 'Test', I would add at the end of the url parameter of the launch: & = MaVariable_D Test

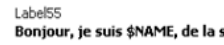
## 5.5 Auto creation of variables.

The auto connection dialog 'global variables, you can automatically create global variables depending on columns in a table. To do this specify the connection and table to use and click 'Auto Create Variables'.

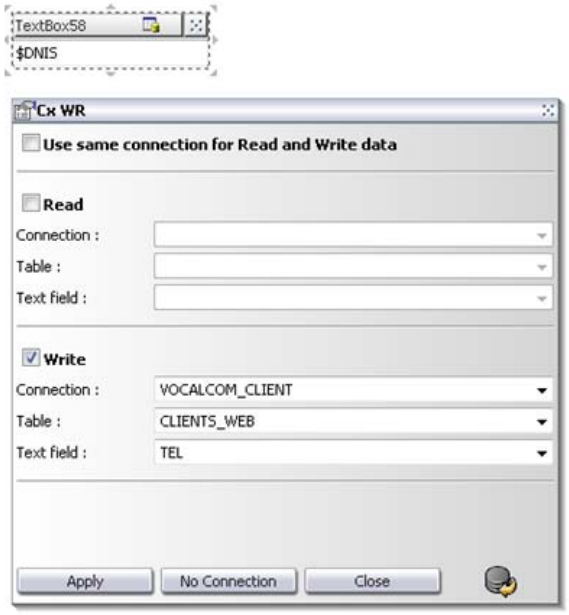
## 5.6 Utilisation des variables.

Global variables can be used at different places in a script. They can be used to assess the content of a label or a text field, or any other object having the property 'Evaluate'.
For example (see screen below)

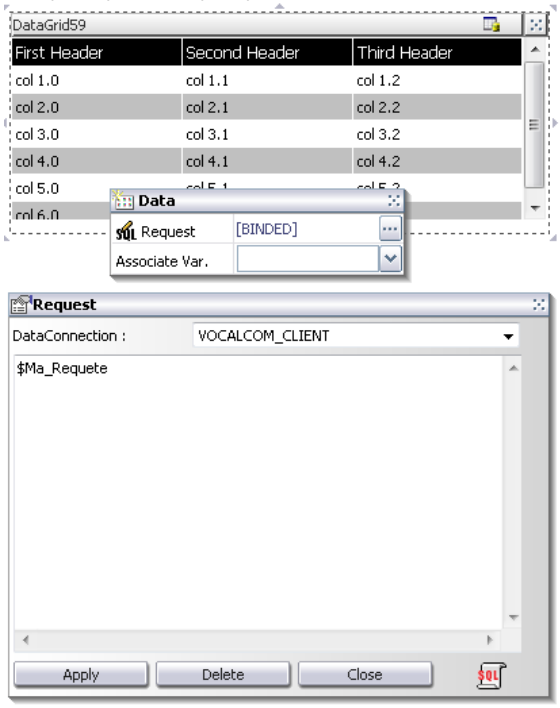
to display the name of the agent in a welcome sentence



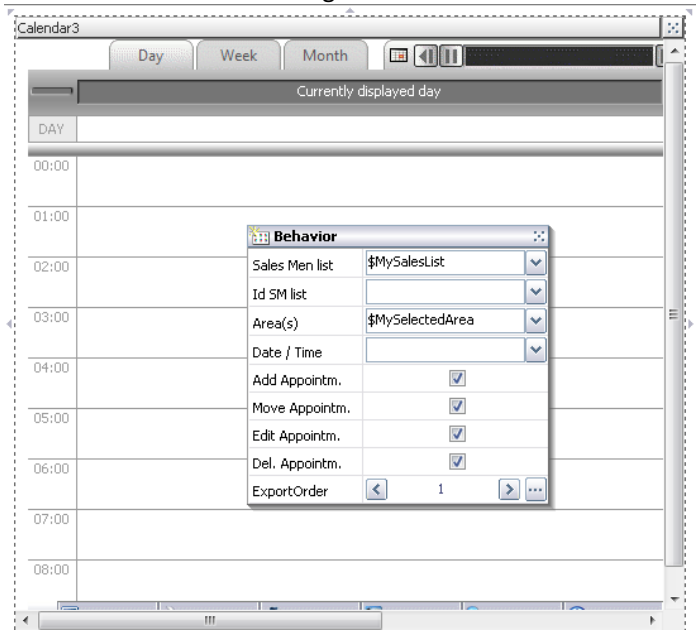

- To store the callers phone number in the database.



- To specify a SQL query.




- To initialise fields from the agenda.



# 6. Appendix « Connections and management of customer tables»

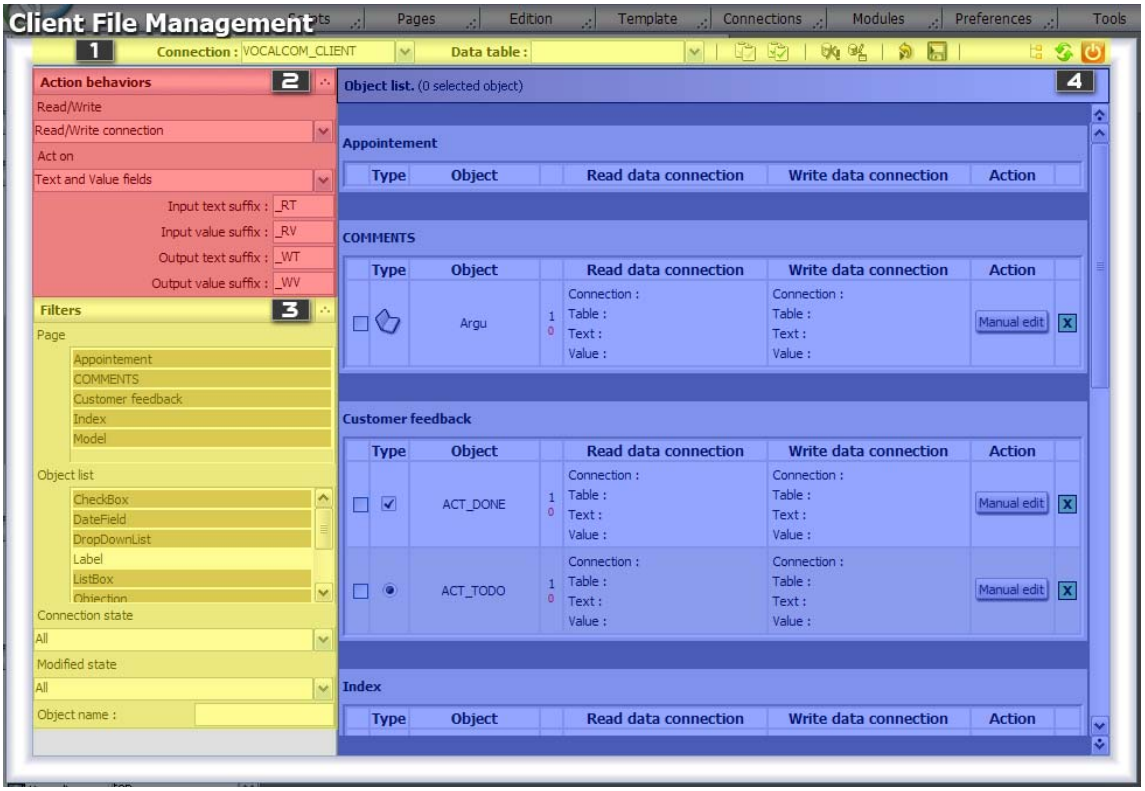
The 'Connections and tables management client can connect automatically to objects in the script fields of databases defined in the Administration Hermes.Net, either automatically by connecting to existing fields or creating a new table corresponding script.

By several criteria the application will search the fields in a table corresponding to the name of Script objects. If the field is found, the object will be automatically logged in, if not a new field will be created automatically in the database.
This module can also be used to disconnect all the objects in the script of the database.

This module is compatible with Microsoft SQL Server and Oracle databases.

By default, the columns will be created of type "nvarchar", accepting a maximum of "255 characters", allowing the values "null" and without default. You can change these values for each field in the base connected to an object.

The connection module includes four areas :




1. This menu is the menu of actions of the module. Via this menu you can select the objects, save the changes ...
2. This area allows you to set the parameters for auto-connection objects.
3. This area allows you to set the various filters on self-connection objects.

4. This area displays the objects in the script and can view their status.

## 6.1 Setting the parameters of auto-connection

The parameters of self-connection can define how the columns will be created in the database or how the correspondence between object name and names of columns in a table of a database "will be performed.

The available parameters :

| | |
|---|---|
| Read/write | This parameter specifies whether the objects will be connected:<br>• Reading and writing<br>• Read only<br>• Writing only<br><br>An object connected read / write read data in the connection of "reading" to reload the page and written in connection to "write" on the "registration page" or during browsing.<br><br>An object can be connected read / write to two different fields, or the same field. Similarly, an object can be connected for example in writing. |
| Act on | This parameter specifies if the connection is self-acting:<br>• The text and the value of the object<br>• The text of the object only<br>• On the value of the object only<br><br>This setting only applies to objects with a text and a value.<br>It is possible to connect and the text and the value of the subject to the same fields, though the value of an object is often used for reading and storage of "code" while the text is used to display user.<br><br>It should also be noted that if an object is connected to read a column for text and a column for the value, if both fields have a different value, the value will be used.<br><br>It is not possible to connect the text and the value of an object at two different tables in a database. |
| Column suffix text entry | This parameter lets you specify a suffix that is automatically added to the name of the column created to connect to the "text" of an object "read".<br><br>This field can be empty if you do want to add a suffix to the name of the column. |
| Suffix column value entry | This parameter lets you specify a suffix that is automatically added to the name of the column created for the connection to the "value" of an object "read".<br><br>This field can be empty if you do want to add a suffix to the name of the column. |


Vocalcom - 7 rue de Tilsitt – 75017 Paris - Tél. : +33 1 55 37 30 50 – Fax : +33 1 55 37 30 51 – www.vocalcom.com
SA au capital de 5 000 000 € – RCS Paris B 401 973 631 – TVA Intracommunautaire : FR 27401973631

| | |
|---|---|
| Suffix column text output | This parameter lets you specify a suffix that is automatically added to the name of the column created to connect to the "text" of an object in "writing".<br><br>This field can be empty if you do want to add a suffix to the name of the column. |
| Suffix column value output | This parameter lets you specify a suffix that is automatically added to the name of the column created for the connection to the "value" of an object in "writing".<br><br>This field can be empty if you do want to add a suffix to the name of the column. |

These settings do not carry out "direct action" on the connection module. But they modify the actions that will be made during the validation of auto-connection via the menu "actions».

## 6.2 Setting the auto-connection filter

The auto-connections filters can filter the items on which the module will act. The filter is visible in the objects to be connected. In fact, all objects not corresponding to the filter will be masked.

If a filter mask a selected object, the object is automatically deselected. This ensures that, before the validation of auto-login the user sees the objects on which it will act.

When the user clicks the button to select all objects, only those shown (ie not filtered) are actually selected.

The filters are available:

| | |
|---|---|
| Page | This filter displays the list of all pages of the script.<br><br>If the page is selected in the list (brown background) objects of the page will be visible. In contrast, if the page is not selected in the list (white), all items in this page will be hidden.<br><br>Note: By default all pages of the script are selected. |
| Objects list | This filter displays the list of all the types of objects connected.<br><br>If an object type is selected (brown background), the objects of this type will be visible. In contrast, if an object type is not selected, all objects of this type will be masked.<br><br>Note: By default all types of objects are selected except the type label. |
| Connexion status | This filter allows you to choose whether it wishes to act on all objects regardless of their connection, only the objects already connected to a database (eg disconnect) or only the objects that are not yet connected to the base.<br>The choice is made through a selection list. |
| Change status | This filter allows you to choose whether it wishes to act on all objects regardless of their state of modification, only the objects whose state is changed (a connection had not validated on these objects) or only the |

| | |
|---|---|
| | objects the state is not changed (no action has had on these items). |
| Object names | This field allows you to enter a filter with respect to the name of the object in the script.<br>Therefore, the user must enter the beginning of the desired name. It can use the "*" as a wildcard (any character) anywhere in the seizure.<br>The filter is not case-sensitive characters. |

## 6.3 Explanation of the menu "Actions"

This menu allows you to perform different actions for connecting to client file.

| | |
|---|---|
| Drop-down list "connection" | This list displays the available connections that were created in the Administration Hermes.Net.<br><br>The connection module acts as a connection / table at a time. As against it is possible to perform several actions on each of connections / different tables without having to quit and relaunch the module.<br><br>Before any operation, the user must ensure that it acts on the desired connection. |
| Dropdown list " data table " | This list displays the tables available in the selected connection.<br><br>Furthermore, this list is editable, ie instead of selecting an existing table, the user can specify a new table name that will be created during registration.<br><br>If the user enters the name of a table that already exists in the connection, the corresponding table will be automatically selected.<br><br>The names of the tables will be created automatically capitalized. If the user enters a name in lowercase, it will be converted automatically. |
| Bouton « tout désélectionner » | This button allows you to deselect all currently selected objects. |
| Bouton « tout sélectionner » | This button will select all visible objects (objects are not filtered selected). |
| Bouton « tout déconnecter » | This button allows you to disconnect all the selected objects from the database.<br><br>The changes will be effective after registration.<br><br>If no object is selected, the button is disabled. |
| Bouton « tout connecter » | This button allows you to connect all the selected objects to the database. |


Vocalcom - 7 rue de Tilsitt – 75017 Paris - Tél. : +33 1 55 37 30 50 – Fax : +33 1 55 37 30 51 – www.vocalcom.com
SA au capital de 5 000 000 € – RCS Paris B 401 973 631 – TVA Intracommunautaire : FR 27401973631

| | |
|---|---|
| | If fields in the table selected match the name of an object (in agreement with the parameters entered self-connection), it will be connected to existing fields. Otherwise a new field will be added to the table.<br><br>The changes will be effective after registration.<br><br>If no object is selected, the button is disabled. |
| Bouton « tout réinitialiser » | This button to reset the changes not yet saved. |
| Bouton « Enregistrer » | This button allows you to save the changes.<br><br>If there is a change summary window will be displayed to indicate the operations to be performed.<br><br>If there is no change, a message will indicate that there is nothing to save. |
| Bouton « Vérification des connexions » | This button allows you to launch the application for verification of connections in a new window. |
| Bouton « Recharger » | This button allows you to restart the application for connection and management of customer tables.<br><br>If changes are not saved, a message will be recorded before reloading. |
| Bouton « Quitter » | This button will quit the connection and client management table.<br><br>If changes are not saved, a message will be recorded before reloading. |

## 6.4 Management of objects to connect

### 6.4.1 General interface

This area allows you to manage and view the objects to be connected.

The objects are organized by page. If a page is filtered, it will not be visible in the interface.
All unfiltered pages are displayed in the interface. They are displayed in order to collect the objects they contain. It is possible to navigate from page to page through the buttons and respectively at the top and base of the scroll bar. The pages are displayed in aliphatic order.

1. At the top of the interface language indicates the number of items currently selected.

   Within a page the interface displays the items line by line, sorted alphabetically by object name.

   The information of an object is provided in a column, in order:
   1. Check box to select the object. For the self-connection acts on the object must be selected.
   2. Image indicating the type of object.

3. Name of the object.
4. Indicators of occurrence. The column will display indicators of occurrence to see quickly if several fields in a database likely to be connected to objects of a different script. Two indicators are provided (one per line):
a. This number (in black) shows the number of objects of the same name in the script (whatever filters selected).
b. This number (in red) indicates the number of objects of the same name in the currently selected script.

If both numbers have the same value, this means that objects in the script with the same name will be connected to the same field from the database.

5. Login information in reading. This information will be detailed in the next section.
6. Login information in writing. This information will be detailed in the next section.
7. This button allows you to manually edit the connection of the object.
If the connection is being manual for the item, the button is renamed to "Apply".
8. This column allows you to view the current state of changing the subject

| | |
|---|---|
| X | Indique que les connexions de l'objet n'ont pas été modifiées depuis la dernière sauvegarde. |
| X | Indique que les connexions de l'objet ont été modifiées depuis la dernière sauvegarde. |

### 6.4.2 Details on connection indicators reading and writing

Indicators of a connection object is given in columns "Connecting Reading and Writing Connection".

| Read data connection | | | Write data connection |
|---|---|---|---|
| Connection : | VOCALCOM_CLIENT | | Connection : |
| Table : | CUSTOMERS | | Table : |
| Text : | | | Text : |
| Value : | Argu_RV | 1 | Value : |

In these columns a complete description of the connection of the object is displayed on four lines to fill in:

1. Connection: connection name where the object is connected. This name corresponds to the description of connection that has been entered in the Administration Hermes.Net.
If the object is not connected, there is no value given.
2. Table: table name where the object is connected.
If the object is not connected, there is no value given.
3. Text: column of the table where the object is connected to the "text".
If the object is not connected, there is no value given.
Opposite the name of the column number indicates the number of other objects connected to the same column (for the text or value).
4. Value column of the table where the object is connected to the "value".
If the object is not connected, there is no value given.
If the object has no value (only one connection to the text), the indicator "---" is shown in front of the field value.
Opposite the name of the column number indicates the number of other objects connected to the same column (for the text or value)

If the method for manual editing is not enabled, this information is textual. For against the edition if the hand is turned to the subject, the text are replaced by selection list.

The connection information may be entered in black or red.
The color red does not mean that there is an error, but can indicate that the connection / table of the object does not correspond to the connection / table selected in the Action menu.
This indication provides a quick way to identify objects connected to an old connection that the user might want to replace.

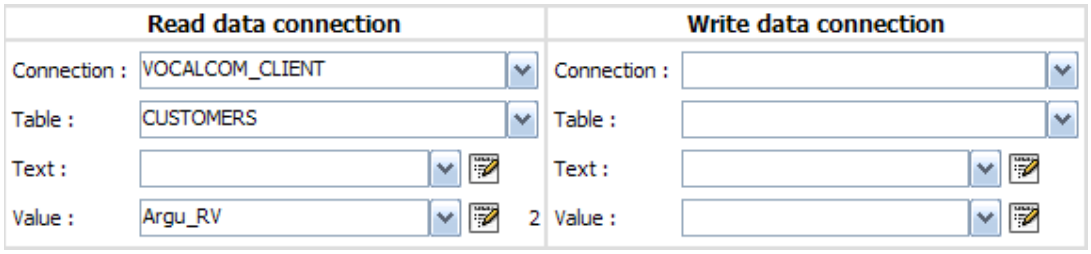
On the screenshot we see that the object is connected in reading but not writing.

### 6.4.3 Details on the manual editing of connections

The manual editing allows you to connect the object without using the engine to connect automatically. This mode can also change the properties of a column in the database.

If the user clicks the button "register" all the fields being edited manually will be automatically validated.

When the manual mode is activated the connection of textual information of the object are replaced by lists :




1. Connection: Allows you to select a connection to a database that was created in the Administration Hermes.Net.
2. Table: Allows you to select a table from the selected connection.
You can enter the name of a table that does not currently exist. In this case, the table will be created (when saving changes). The table names will be automatically converted to uppercase.
3. Text: Allows you to select a column of the table selected.
You can enter the name of a column that does not currently exist. In this case, the column will be created (when saving changes).
4. Value: You can select a column of the table selected.
You can enter the name of a column that does not currently exist. In this case, the column will be created (when saving changes).

If an object is already connected, the lists are automatically positioned on the corresponding values.

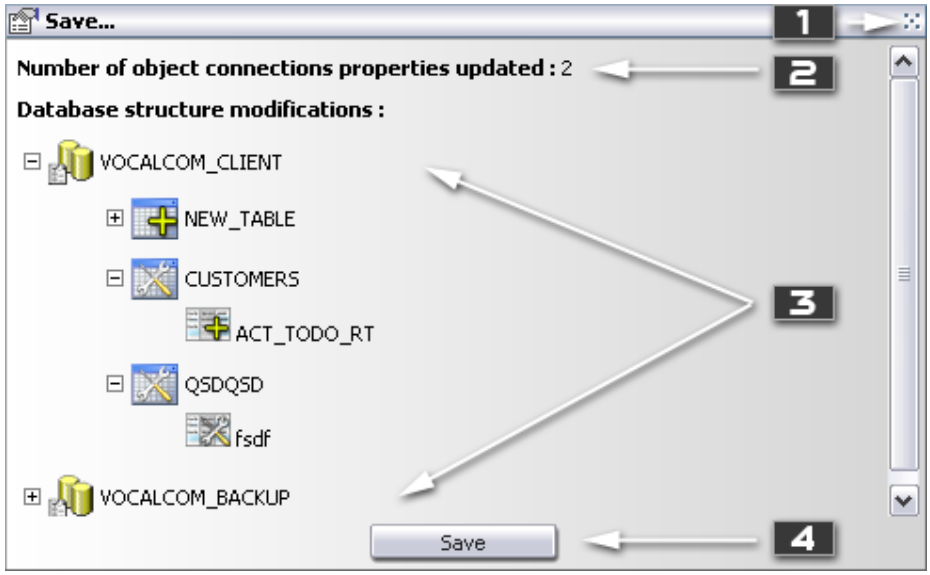
If an object is already connected and the user exchange connection in the list, the lists of "table", "text" and "value" is automatically reset to allow the user to select the correct information.

The button to the right of this selection lists for the text and value can change the properties of the column in the database:

1. This language allows to know the login / table / column changes will be applied.
2. This list can change the type of the column. Possible values are:
a. nvarchar: the text column accepting Unicode characters that can contain a maximum of 4000 characters.
b. ntext: column text accepting Unicode characters that can contain a maximum of 1073741823 characters.
v. int: column can store whole numbers ranging from - 2147483648 to 2147483647.
3. Size of the column.
If the type is selected other than "nvarchar", this field will be disabled and the default value will be automatically entered.
For fields of type "nvarchar" the maximum size allowed by this module is "499." Beyond this size we should create fields of type ntext.
4. Check box indicating whether the column allows or not the value "null" (no value). If you do not want the column allows the values "null" you must add a default value.
5. Default value of the column.
The value entered can not be larger than the size specified for the column (if available, an error message will be displayed during the validation of these parameters).
If the column is of type "int" default should be a whole number (if available, an error message will be displayed during the validation of these parameters).
If a default is specified then the column is configured to accept the value "null", the default value will be ignored.
6. This button is used to validate the changes to the column.
7. This button allows you to close the window property on the column without validating the changes.

## 6.5 Registration Window

When the user clicks the button "save" a window indicates the number of objects whose information is changed and the summary of transactions that will be made on the structure of databases.
The tree displays the objects on which amendments are made.



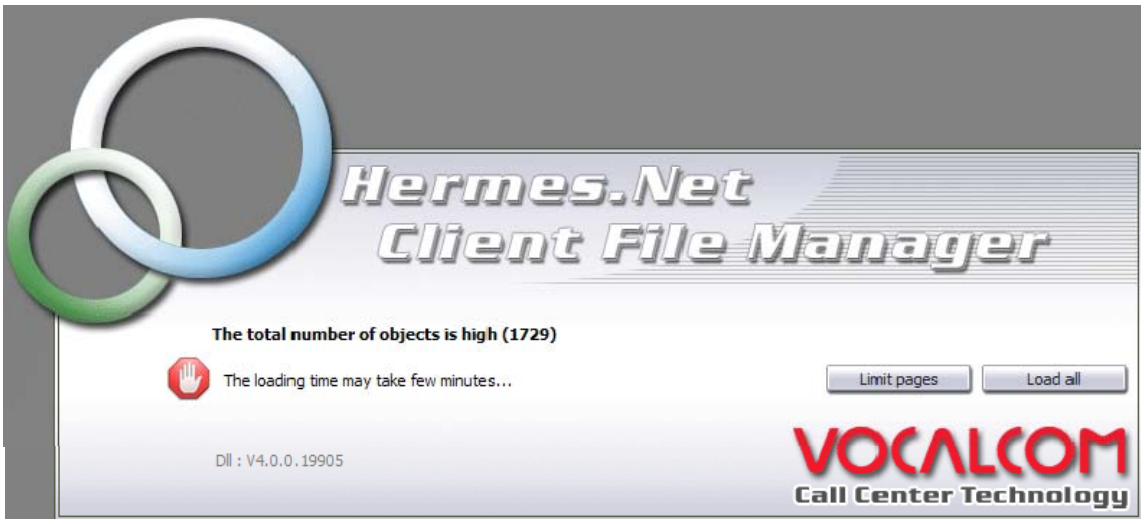

1. To close the window to cancel the registration.
2. Indicates the number of items which the login information has been changed.
3. Displays a tree that shows the changes that will be made on the structure of databases.
The tree is organized by tables  columns.◊ ◊connections
4. Used to validate the changes.

The tree of changes to identify visually the type of changes to be made on the structure of databases :

| | |
|---|---|
| | This icon indicates a connection in the database. A table of the connection will be changed, or a table will be added to the connection. |
| | This icon indicates that the table will be created. |
| | This icon indicates that the table will be changed: a column will be changed in the table, or column will be added to the Table. |
| | This icon indicates that the column will be created. |
| | This icon indicates that the properties of the column will be modified. |

## 6.6 Window limiting the number of pages

When launching the module connection, you can see the display window for limiting the number of pages:




Indeed, the number of objects in the script, the greater the load time and processing the application will be important (the self-connection, selection of objects, ...).
The application displays this message if the number of objects in the script is more than two hundred.

If you want to perform a comprehensive whole script, you can load all objects of all pages by clicking on the "All tasks".
As against if you do want to connect the items to specific pages by clicking on the button "Limit the pages you can choose to display the pages in the application.
The user can also choose to connect the objects of all the pages and run the application several times, each time on different pages which can be a time saver if the object is really important.

If the user chooses to change the number of page, a popup allowing him to choose to display the pages will be displayed. This popup is composed of a multiple selection list (Ctrl + click to select multiple pages) and a button to validate the selection.

## 6.7 Examples of uses

### 6.7.1 Using similar to the application for connection to the client file Scripter v3.

To get the same result in creation of client file with the module "Creating the client file from the script" Script V3, set each parameter as follows:

1. Section behavior:
• Read / Write: leave the default which is "Connect Read / Write"
• Influencing: leave the default which is "text and data fields"
• Suffix column txt reading: putting the empty field
• Suffix column val read: put "_VAL" in the field
• Suffix txt column writing: to the field to empty
• Suffix column val write: put "_VAL" in the field
2. Section Filter:
• Page: leave all the selected pages (default)
• List of objects: select the "Label"
• State connection: select "Objects are not connected"
• Status change: Leave the default "All".
• object name: leave the default (not completed).

3. Menu bar:
1. Select the connection in which you want to add the new table.
2. Enter the name of the table you want to create.
3. Click the button "select all" to select all items not filtered.
4. Click on "connect everything" in order to auto-connect all the objects
5. Click "Save" to save changes

### 6.7.2 Disconnecting all the fields in the script

This module can be used to disconnect all the objects in a script database. This can be done in a few steps:

1. In the filters, check that all objects are displayed (the default type objects "Label" are filtered out).
2. Click the button "select all".
3. Click the "all out".
4. Click "Save."

It is important to note that this module disconnects the objects connected to the base for the recovery and automatic backup of the text and the value :

- Connections and Query objects of type "array of data" will not be deleted.
- The dynamic data objects "drop down" and "multi selection list" will not be removed.

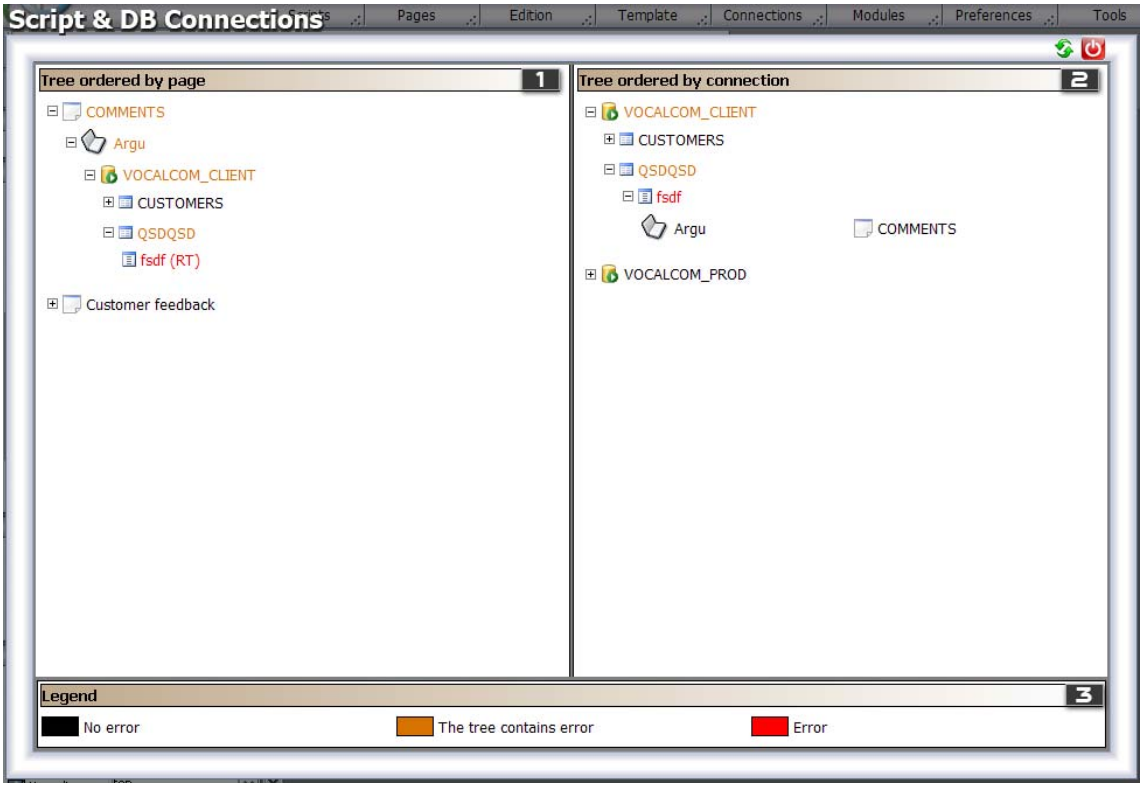
### 6.7.3 Modification of all existing connections of the script

To change all existing connections of the script, you can follow the following steps:
1. Section behavior: the behavior set of connection desired.
2. Section filter: Make sure all objects are displayed (the default type objects "Label" are filtered out).
3. In the Action menu, select the new connection from the script.
4. In the table menu, select the new table of the script (existing table or a table to be created).
5. Click the button "select all".
6. Click on "connect everything".
Note here that there is no need to start by disconnecting all the objects. Indeed, an object that is already connected will be amended if its connection is selected.
7. Click "Save".

# 7. Annex "Checking connections"

The verification module connections can view connections to databases made in the script and especially the mistakes he might find (or table fields that no longer exist, for example).

This module does not change the connections of the objects or the structure of databases, but provides a quick way to detect problems on the connections of the script.




The interface module is separated horizontally into two distinct areas:
1. Left tree displays different pages of script in which objects are connected.
2. On the right tree shows the different connections in which objects are connected.

The third area of the interface displays the legend to the user.

For both trees the same color code is used to detect errors:
• Black text: there is no error in the tree.
• Text orange tree contains errors. By opening branches it will be possible to identify the problem.
• red text: the error is at this level.
If the error is on the name of the connection: the connection has been removed from the Administration Hermes.Net or the database is inaccessible.
If the error is in the name of the table: the table was deleted from the database.
If the error is in the name of the column: the column was removed from the table.

The left tree (tree organized by page) is structured as follows :

- Name of the page[1]
  - Name of the object in the page[2]
    - Name of connection used
      - Name of the table used
        - Name of the column used

Opposite the name of the column indicates a code for which data object is connected to this column:
• RT: the field is connected to the text reading
• WT: the field is connected to the text in writing
• RV: the field is connected to the reading
• WV: the field is connected to the write

The right tree (tree organized by connection) is structured as follows:

- Connection name[3]
  - Table name[4]
    - Column name[5]
      - Name of connected object / name of the page where the connected object is placed

# 8. Appendix "Query Editor"

The 'Query' is a graphical tool to easily create queries on a database.

This module is compatible with databases like Microsoft SQL Server and databases like Oracle.

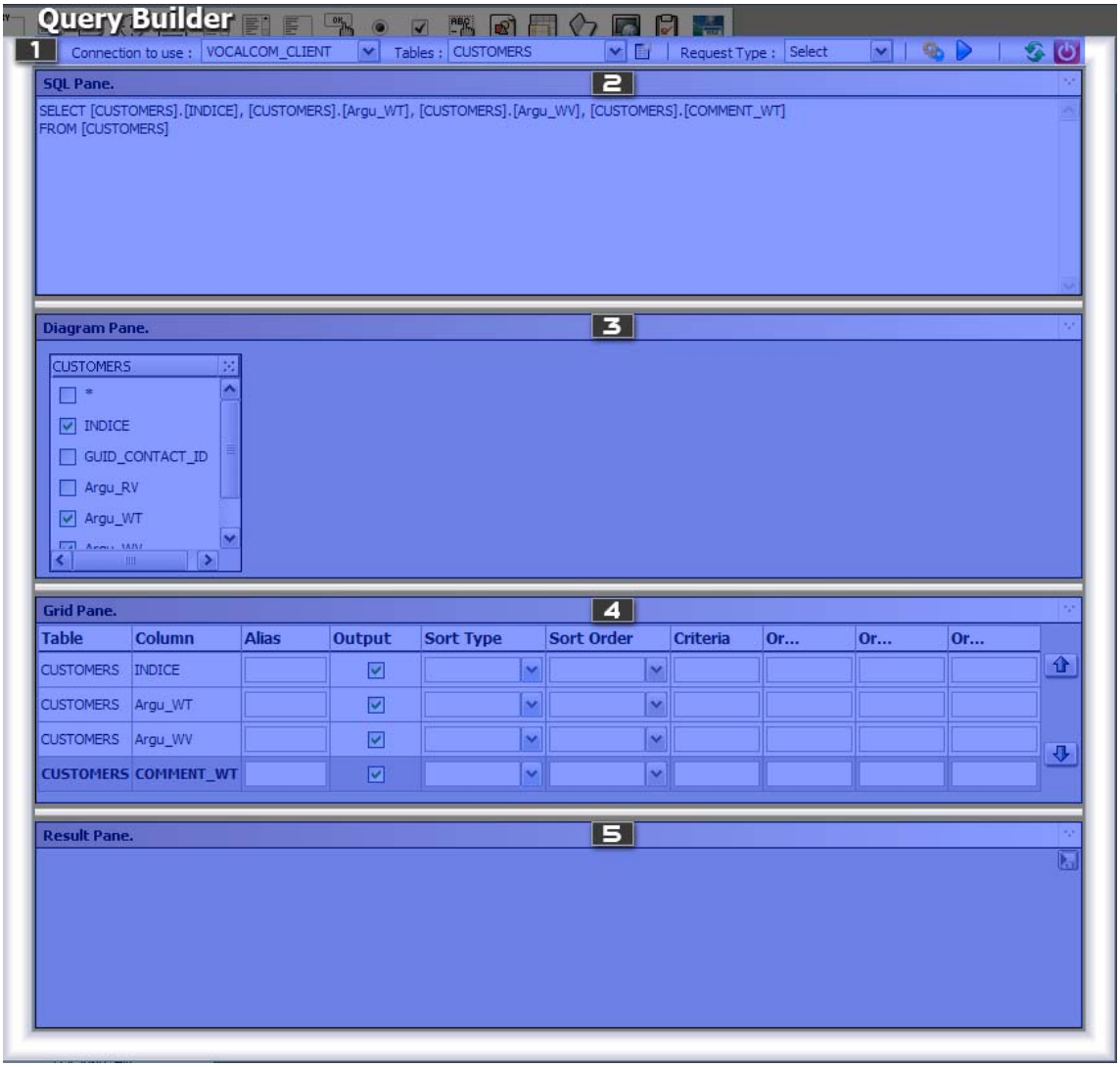
## 8.1 Basics

The module editing application may be launched with an existing application or not.
When it is launched with a request, the interface is automatically initialized with it. Otherwise, the interface is launched "virgin".

A request may be made only on a database connection at a time: this module does not support creating query on different servers linked.

The interface of the query is composed of five main areas:

[1] Only pages containing articles connected to a database are present in the tree.
[2] Only items connected to a database are present in the tree.
[3] Only connections that are connected objects are displayed in the tree.
[4] Only tables which are connected objects are displayed in the tree.
[5] Only columns in which items are connected are displayed in the tree.




1. The first area is the menu module, it provides access to settings and activities available.
2. The second box allows you to view and edit the SQL code of the application.
3. The third area will display the diagram of the tables involved in the request and the relationship between these tables.
4. The fourth area to display the grid of columns involved in the complaint and thus create the conditions, and the order of result set.
5. The fifth area will display the result of query execution.

A horizontal bar separates zones 2 / 3, Zone 3 / 4 zones and 4 / 5. These bars allow you to resize the boxes simply by moving the bar vertically with the mouse.

Each zone (except the menu action) is composed of two parts: the title of the area and its contents. The button to the right of the title bar will minimize / maximize the area. Only the title bar is visible to a minimized.

In all cases, the user can edit or write text directly into the query box displays the SQL code. If the user changes the text in this area, other areas will be updated automatically when the user clicks on the interface to validate its amendment.

In contrast, if the user changes the query from the graphical tools available, the SQL code is automatically modified.

### 8.1.1 Elements list of the query editor

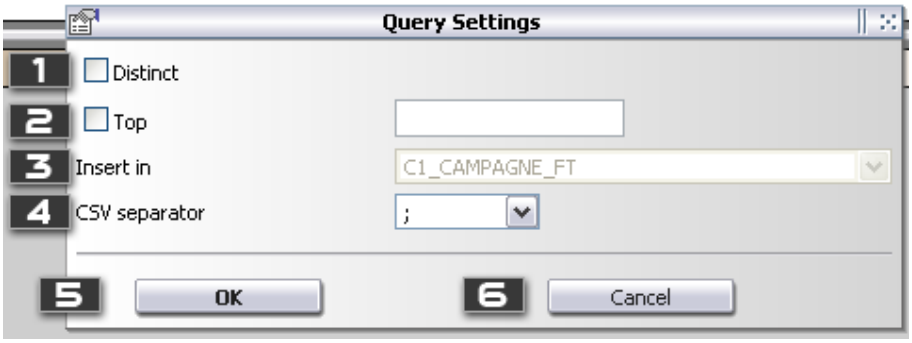
The menu module is composed of the following:

| | |
|---|---|
| Connexion to use | This list lets you choose the connection to be used to create the query.<br>If the editor is opened with an existing query, the list screens the connection used by the query.<br><br>Before anything else the user must select the connection on which to edit the application (available connections are connections that have been created in the Administration Hermes.Net).<br><br>If the user changes its connection while a complaint is issued, the application will display a confirmation message indicating that it continues the current request will be deleted. |
| List of connexion tables | This list displays the tables of the connection that is selected.<br>If no connection is selected, this list will be empty.<br><br>To add a table to the application you must select from this list. |
| Add the table to the query | This button allows you to add the selected table to the query.<br>If no table has been selected, this action will display the message "Please first select a table». |
| Choosing the type of query | This list allows you to select the type of request.<br>The types available are:<br>▪ select (type of request « select from »)<br>▪ insert from (type of request « insert from »)<br>▪ insert into (type of request « insert into »)<br>▪ update (type of request « update »)<br>▪ delete (type of request « delete from »)<br><br>If the user changes the type of motion when a complaint is issued, the application will display a confirmation message indicating that it continues the current request will be deleted. |
| Request parameters | This button allows you to display the parameter of the request.<br>This window is shown in the following section. |
| Request execution | This button allows you to run the query.<br>If no request has been created, this button will display the message "Please first create a query". |
| Reload the quesry editor | This button allows you to reload the query.<br><br>If the editor was opened with a query and it has been modified, a message will ask the user whether to save the new application in |

| | |
|---|---|
| | before calling reload the interface. The user can then save it, do not save or cancel the operation "reload page"<br>. |
| Exit the query editor | This button will quit the query editor.<br><br>If the editor was opened with a query and it has been modified, a message will ask the user whether to save the new query from the caller before leaving the interface. The user can then save it, do not save or cancel the operation "quit". |

### 8.1.2 Window parameter of the request

This window allows you to choose the parameters of the query. This window is accessible via the button "Parameters of the request" from the menu bar.
When the user selects the type of query "Insert", this window opens automatically to allow the table to choose where to insert data.




1. Allows you to specify that the request should return the results without redundancy
2. Allows you to specify the maximum number of lines that must return the application. To activate this mode, tick the checkbox and enter the maximum number of lines in the text field opposite the check box.
3. Pick list to select which table to insert data for a query "insert". If the request type is not "insert", the selection list will be disabled.
Tables proposed in the list are the connection tables selected in the Action menu. We must therefore choose a connection before you open this window.
4. Selection list to specify the column separator CSV files created.
The list offers separators ";", "" and "tab".
5. This button is used to validate the request and close the window.
6. This button allows you to cancel the changes on the parameters of the request and close the window.

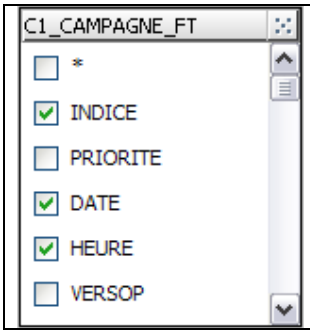
## 8.2 Select query

To create a query for selection, you must first select the connection where the tables on which you want to select data.
Then you add (the) table (s) by clicking the "Add the table to the query" from the menu action.

The table is then automatically added to the diagram. If the table contains one or more primary-key columns, a default query is created that selects the primary key columns of the table. If the table

does not contain a primary key, no query is created.

The selection of columns involved in the request (columns or columns stakeholders results in terms of selection) is done directly in the diagram :



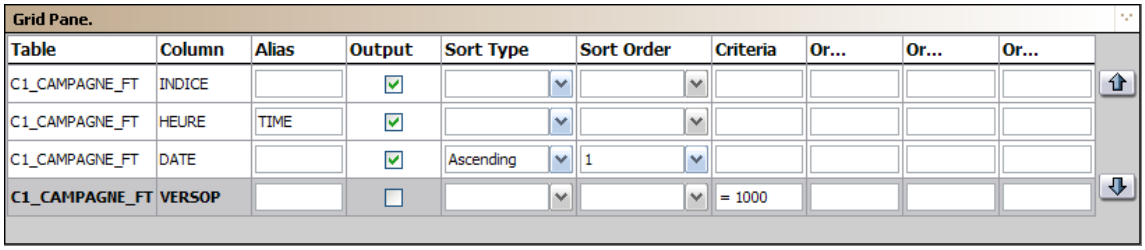

The area chart displays the tables added to the application.
For each table, the diagram shows all the columns preceded by a checkbox to select for the query.

In addition to the columns, a "*" is added which allows to choose a query that returns all fields from the table.

Each column is added automatically added to the grid of columns involved in the request. By default they are selected to be part of the result set of the query.

Attention: on the contrary, if a column is unchecked in the diagram, the column is automatically deleted from the query and the data set on this column (condition, sorting, ...).

The grid columns to adjust the role of the column in the query :




1. This column displays the table belonging to the column (it is possible to create queries using multiple selection tables).
2. This column displays the name of the selected column.
3. This column allows you to specify an alias for the column: the alias is the name of the column in the result set.
If no alias is specified, the name of the column in the result set will be the real name of the column.
If the "*" is specified, it is not possible to choose an alias on this line.
4. This check box lets you specify whether the column will be returned or not in the result set.
5. Selection list to specify whether the result set will be sorted on the column (clause "ORDER BY"). Possible values are:
a. "Empty": the result set is not sorted on this column.
b. "Ascending": the result set is sorted in ascending on this column.
v. "Descending": the result set is sorted in descending on this column.
6. Selection list to specify the sort order for column (scheduling in the clause "ORDER BY").
If the column does not sort the result set, the selection list is disabled.

The list allows to choose an index ranging from "1" to number of columns on which sorting is configured.

The column order is the lowest one used mainly for sorting.

On this example, to sort the results by date and time, we must select the "1" in the column "DATE" and the "2" on the "TIME". Thus the results are sorted by date and results of the same date will be sorted per hour.

When the user selects an order, if another column was already set on this same order, it is automatically changed so that two columns is the same sort order.
1. This column allows you to specify the selection criteria (clause "WHERE").
If the user types the test without the comparison, the "=" will be added automatically. The user can modify if desired directly in the text box.
If the user types an alphanumeric string, the characters " '" will be automatically added at the beginning and end of the chain. Against by the user types a number, they will not even if the column is of type string. In this case, the user must add them.
2. The three columns "or" to specify the alternative conditions of the test ( "OR").

The two arrows on the right of the grid to change the order of the column in the result set: the highest column in the grid will be the first column of result set column when the lowest in the grid will be the last column of the result set.

Attention: the order of columns in the grid affects the construction of terms "AND" and "OR" between columns. The construction of these criteria is detailed in "Creating advanced criteria.

To select the line-on which the order should be changed, simply select it by clicking it. The selected line is subline gray.

## 8.3 Query insert from another table

When the user selects in the list of types of query type "Insert", the window query parameter is automatically displayed to allow the choice of the table where the data will be inserted.
At any time the user can reopen the window parameter to change the table insertion.

The table insert is not added to the diagram. The zone chart is used to specify the tables, a selection will be made to enrich the data in the table insert.

As a request to insert data from another table is used to insert data using a query selection, the setting is close to that of a selection query:
1. Adding tables involved in the query box diagram.
2. Select columns to select or on which there are conditions in the area diagram.
3. Setting the query in the grid columns.

The grid column has a parameter additional to that shown for queries selection: the "Add.

The column "Add" contains a selection list to specify the correspondence column in the table where the data will be inserted.

It is not possible to specify the columns of the table insert, but if the game result of the query selection must return the same number of column as the table insertion.

As against, if you specify the name of a column, it is important to specify the names of all the correspondence columns of the result set
.

## 8.4 Insert query

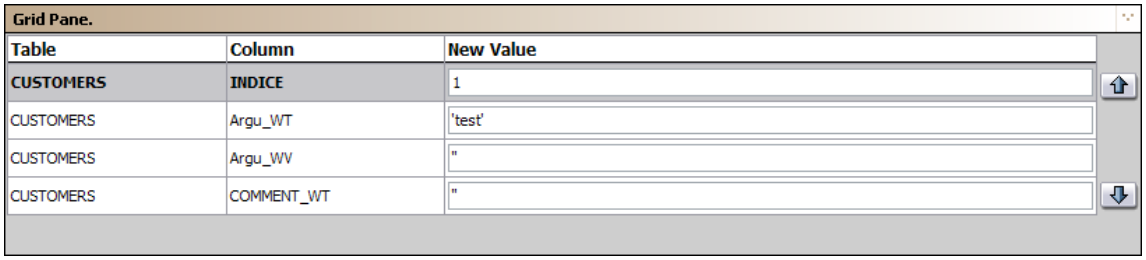
To create a query to insert, you must first select the connection where the table on which you want to insert data.
Then you must add the table by clicking the "Add the table to the query" in the action menu.

An insert query can only be made on a table at a time. If a second table is added, a confirmation message asks if the user really wants to replace the existing table.

To create a request for inclusion, the following steps should be followed:

1. Add the table where to insert the data.
2. Select columns to insert data in the chart (the "*" is disabled).
3. Setting values to be inserted in the grid columns.

The grid of columns in a query insert can specify the values to insert for each column :




1. The first column is a reminder about the name of the table where the data will be inserted.
2. The second column is a reminder on the column name (if the column allows the values "null" or has a default value, it is not required to select in the chart. It is mandatory in cases contrary).
3. The last column allows you to specify the value to insert.
The inclusion of "empty string" must be explicit in specifying the value "''".

If the user types an alphanumeric string, the characters " '" will be automatically added at the beginning and end of the chain. Against by the user types a number, they will not even if the column is of type string. In this case, the user to add.
4. The arrows ordonnancement columns to the right of the grid here have an effect organizational chart but have no impact on the request.

## 8.5 Request update

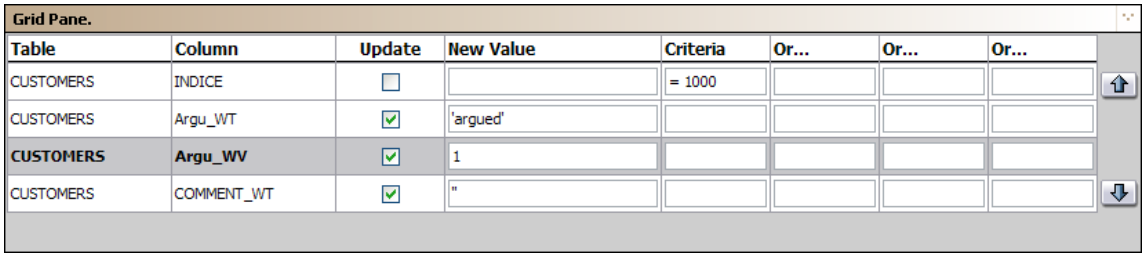
To create a query and update, you must first select the connection where the table on which you want to update the data.
Then you must add the table by clicking the "Add the table to the query" in the action menu.

A request to update can only be made on a table at a time. If a second table is added, a confirmation message asks if the user really wants to replace the existing table.

To create a query and update, the following steps should be followed:

1. Add the table or update data.
2. Selection in the chart where the columns to update the data (the "*" is disabled).
Selection in the chart columns involved in the criteria for the update.
3. Setting values to be inserted in the grid columns.
Setting criteria for updating the grid columns.

The grid of columns in a query and update allows you to specify the values to be updated for each column and the criteria for determining what records will be updated :




1. This column displays the table belonging to the column.
2. This column displays the name of the selected column.
3. This check box lets you specify whether the column will be updated. If the column is not checked, its value will not be updated. It will only be used for establishing the criterion for selection of columns to update.
4. This column allows you to specify the criterion for updating (clause "WHERE").
If the user types the test without the comparison, the "=" will be added automatically. The user can modify if desired directly in the text box.
If the user types an alphanumeric string, the characters " '" will be automatically added at the beginning and end of the chain. Against by the user types a number, they will not even if the column is of type string. In this case, the user to add.
5. The three columns "or" to specify the alternative conditions of the test ( "OR").

The two arrows on the right of the grid to change the order of columns for the construction of the status update. The construction of these criteria is detailed in "Creating advanced criteria.

To select the line-on which the order should be changed, simply select it by clicking it. The selected line is subline gray.

## 8.6 Request delete

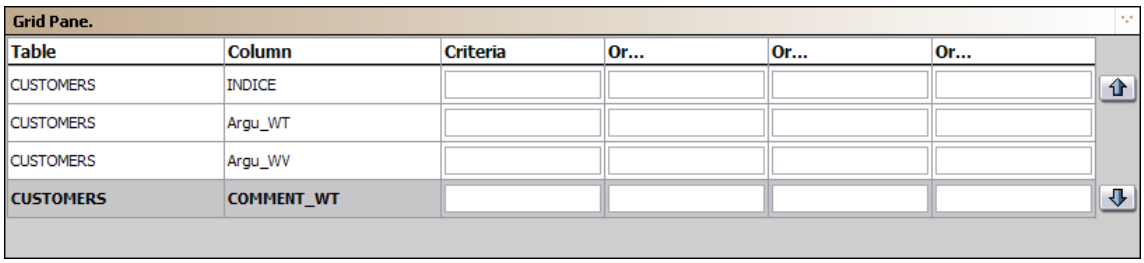
To create a request “delete”, you must first select the connection where the table on which you want to delete the data.
Then you must add the table by clicking the "Add the table to the query" in the action menu.

A request “delete” may be made only on a table at a time. If a second table is added, a confirmation message asks if the user really wants to replace the existing table.

To create a request for removal, the following steps should be followed:

1. Add or delete table data.
2. Selection in the chart columns involved in the criteria for deletion.
3. Setting suppression criteria in the grid columns.

The grid of columns of a suppression motion can specify the criteria for determining what records will be deleted:




1. This column displays the table belonging to the column.
2. This column displays the name of the selected column.
3. This column allows you to specify the criterion for removal (clause "WHERE").
If the user types the test without the comparison, the "=" will be added automatically. The user can modify if desired directly in the text box.
If the user types an alphanumeric string, the characters " ' " will be automatically added at the beginning and end of the chain. Against by the user types a number, they will not even if the column is of type string. In this case, the user must add them.
4. The three columns "or" to specify the alternative conditions of the test ( "OR").

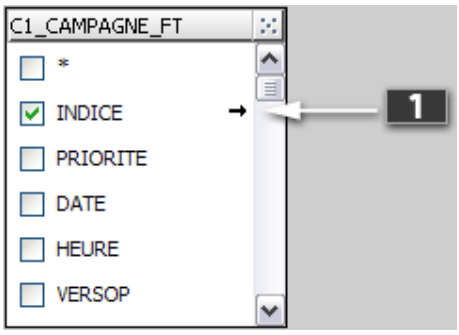
The two arrows on the right of the grid are used to change the order of columns for the construction of the deletion condition. The construction of these criteria is detailed in "Creating advanced criteria.

To select the line-on which the order should be changed, simply select it by clicking it. The selected line is highlighted in gray.

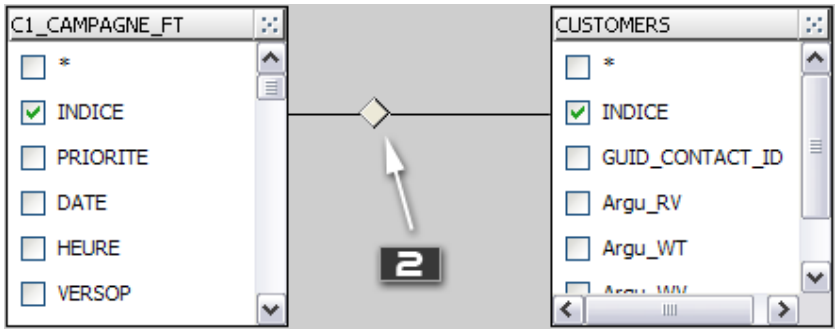
## 8.7 Creating a join between two tables

1. When multiple tables are added to the diagram it is possible to specify the join columns between them.

   To create a join just follow the following steps:
   1. Select the first column of the table where the join should be made. To do this, click on the column in the box diagram.
   An icon indicates that the join is in preparation :

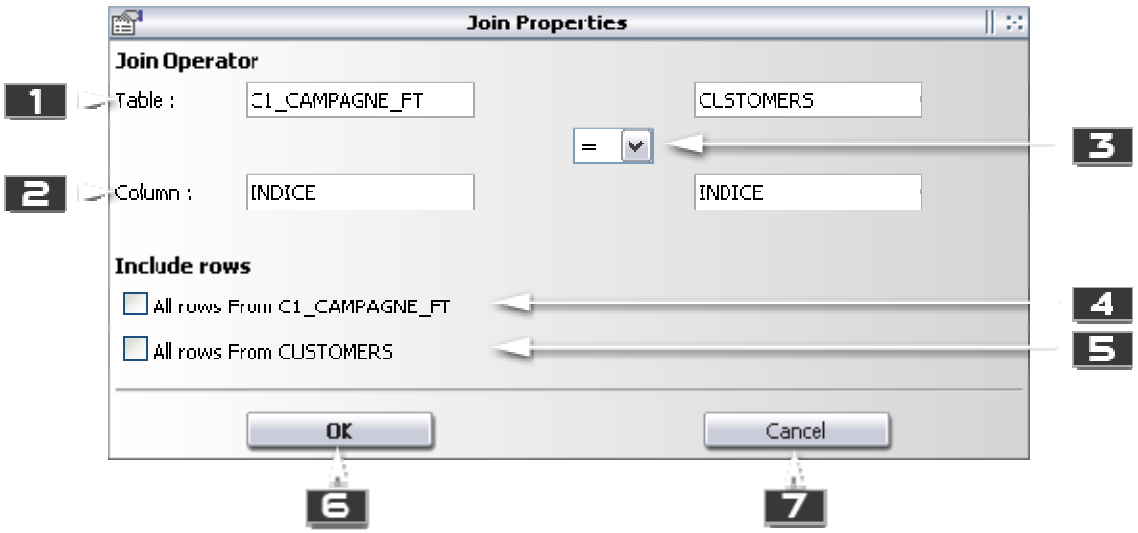



2. Select the second column of the table where the join should be made. To do this, click on the column in the box diagram.
The joint is then drawn on the chart:




By default the joints are created type "INNER JOIN" based on equality of column values (operator =).

To change the properties of the joint, with the mouse right-click the icon to join this will display a context menu. Then select "Edit". The window property of the joints will be displayed :




1. 1. Background on the table in the join. Text boxes are not editable.
   2. Background on the internal column in the join. Text boxes are not editable.
   3. This specifies the operator used for the join between the table / column and the left.
   Possible values are:

a. Equal to (=)
b. Greater than (>)
v. Lower ( "<")
d. Greater than or equal to (> =)
e. Less than or equal to (<=)
f. Different from (<>)

2. Specifies whether all rows of the table must be returned ( "RIGHT OUTER JOIN)
3. Specifies whether all rows from the second table must be returned ( "RIGHT OUTER JOIN and order of tables reversed in the request or" LEFT OUTER JOIN ")
4. Button that allows to validate the parameters of the joint and close the window.
5. Button to cancel the changes on the settings and close the window.
Note: when "4" and "5" are marked, the joint created is of type "FULL OUTER JOIN".

The chart table the request displays a different indicator, depending on the type of join.
At the center of the indicator the operator join is displayed (except in the case of equality where no operator is displayed.

| | |
|---|---|
| | INNER JOIN |
| | FULL OUTER JOIN |
| | LEFT OUTER JOIN |
| | RIGHT OUTER JOIN |

A joint may be removed by opening the context menu (right click on the image of the joint) and selecting "Delete."

If the table is added to the diagram several times tables, a join-type "INNER JOIN" will be automatically created between the two tables on the primary-key columns.

If a table added to the digraph contains foreign keys on a table already present in the diagram, a join-type "INNER JOIN" is automatically created between the two tables on foreign key columns.

## 8.8 Creation of advanced criteria

The grid column of the query allows to create the criteria of the request (for requests supporting criteria).

The grid has four columns concern the criteria: "Criteria" and three columns "or".

If the criterion is entered in the test, it will be like "AND" while in the columns "Or", it will type "OR".

On the same column it is possible to create a condition type "AND".

If the test is done on a single column, that "Criterion" and "Or" is the test will be:
"WHERE (INDEX = 1) OR (INDEX = 2)"

If a test-type "AND" is added to another column of the table, the test will be to type "AND" between these two columns:
CONDITION_COL 1 AND CONDITION_COL 2

Conditions on a column which may include criteria such as "OR".


Vocalcom - 7 rue de Tilsitt – 75017 Paris - Tél. : +33 1 55 37 30 50 – Fax : +33 1 55 37 30 51 – www.vocalcom.com
SA au capital de 5 000 000 € – RCS Paris B 401 973 631 – TVA Intracommunautaire : FR 27401973631

If a type test "OR" is added to another table column (except the first in the grid), without the column "test" is met, the criterion between the two columns will be type "OR"
CONDITION_COL1 OR CONDITION_COL2

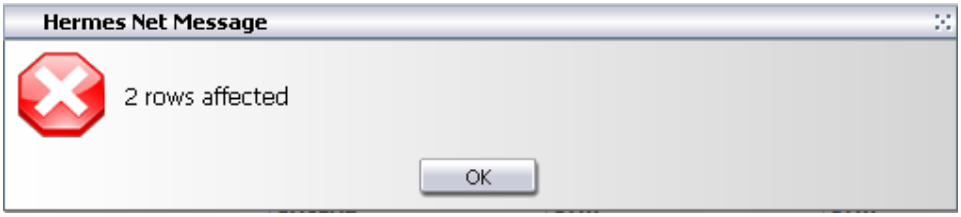
It should be noted that the criteria between columns are not automatically placed in brackets. It is also possible to place the brackets directly into the SQL edit box. These will be kept when editing.

## 8.9 Executing the query

It is possible to execute the query by clicking the button "Run Query" from the menu bar.

If the request is a request for selection, the returned rows will be displayed in the result. In order not to increase the display, only the first fifty lines are shown.
Values of type 'NULL' is shown with the string "<NULL>. The binary type values are indicated with the string "<BINARY>.
Apart from the result grid, button allows downloading the CSV file containing all the selected lines.

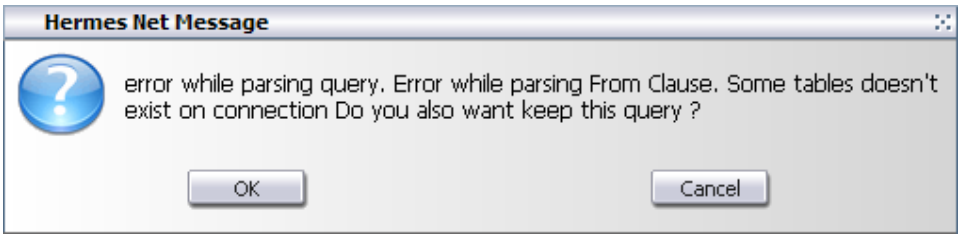
If the request is not an application for selection, after implementation, a window displays the number of rows in the database affected by the query :




In any case, if the query contains errors, a message detailing the error will be displayed after the execution.

## 8.10 Frequent Error Messages when editing existing queries

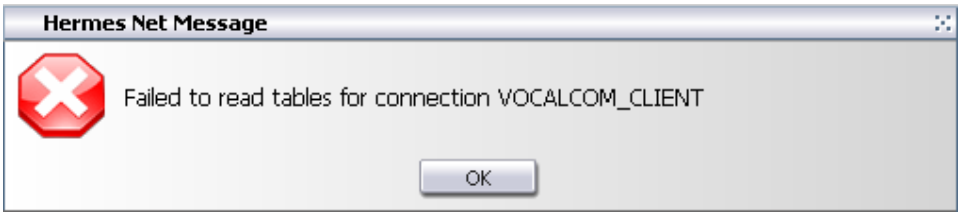
At the opening of the module with an existing application it is possible that errors are detected. For example, if the request is made on a table that does not exist in the selected connection, the module will display the following message:


Hermes Net Message
error while parsing query. Error while parsing From Clause. Some tables doesn't exist on connection Do you also want keep this query ?
OK
Cancel

If the user selects "ok", the request will be displayed in the SQL Editor, but as long as it contains errors other areas will not be fulfilled.
If the user selects "cancel", no queries will be added to the publisher and the user will have to recreate the query.

The text of the error specifies the problem detected. The problem may lie in:
• A table that does not exist in the specified connection.
• A column that does not exist in the specified table.

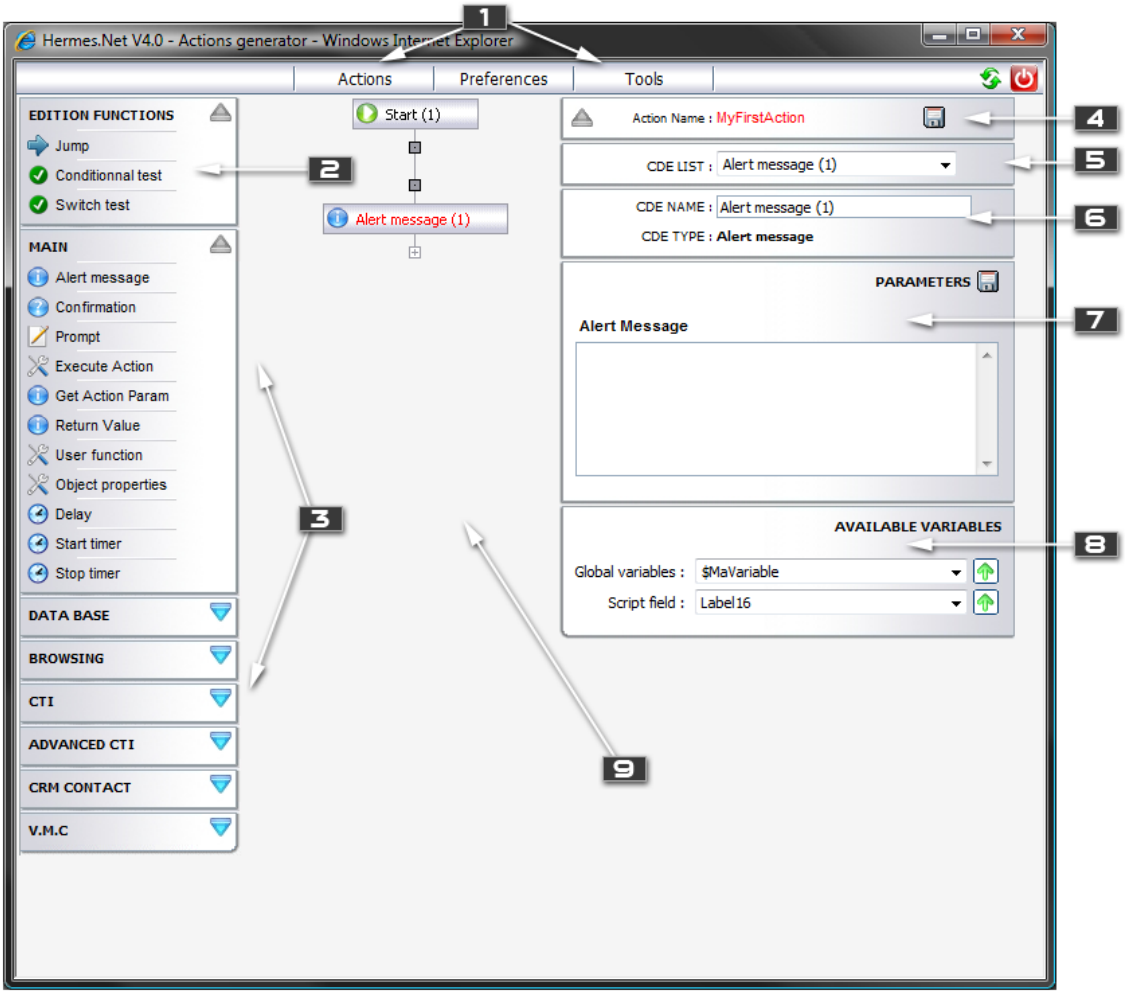
If the connection is specified for the query no longer exists (deleted from the Administration Hermes.Net) or not available (off base server for example), the query will display the message:




In this case, the request will not be updated.

# 9. Annex « Action Builder "

## 9.1 The interface



**9.1.1 Menus. Provides access to various functions of the editor.**

**9.1.2 Orders editions. Allows to place on the working area of the navigation commands, testing, and verification.**

**9.1.3 Orders of shares. Puts in the work zone orders.**

**9.1.4 Displaying the name of the action. The disk icon allows you to save the action.**

**9.1.5 display area from the list of commands present in the action. Select a command from the list to display the settings for it.**

**9.1.6 Viewing and editing the name of the command. Each order can be customized with a specific command name. Under the command name is the type of command.**

**9.1.7 Viewing and editing the list of parameters corresponding to the selected action.**

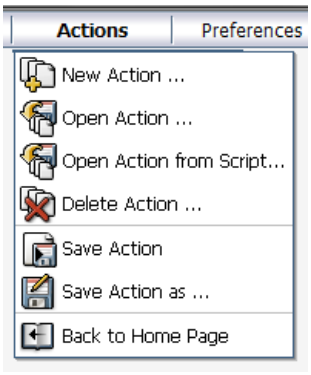
**9.1.8 Display of various types of global variables available depending on the command.**

**9.1.9 Work area. Commands selected in the left menu are placed in this area.**


Vocalcom - 7 rue de Tilsitt – 75017 Paris - Tél. : +33 1 55 37 30 50 – Fax : +33 1 55 37 30 51 – www.vocalcom.com
SA au capital de 5 000 000 € – RCS Paris B 401 973 631 – TVA Intracommunautaire : FR 27401973631

## 9.2 Les menus

### 9.2.1 Menu Actions




**New Action …**
Allows you to create a new Action. If an action is already loaded, a confirmation is requested backup.

**Open Action …**
Can load an existing. If an action is already loaded, a confirmation is requested backup.

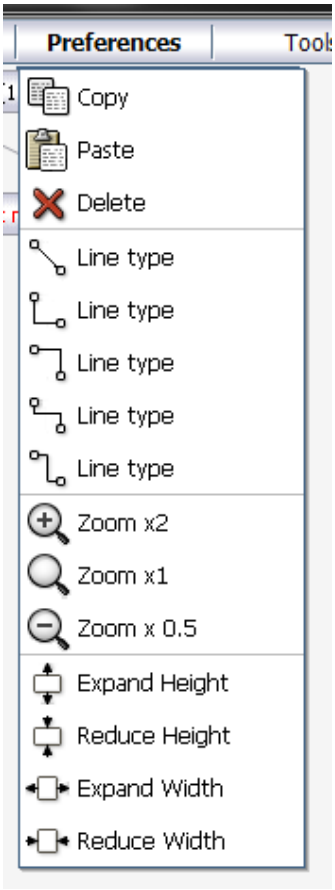
**Open Action from script…**
Can load a script from another. If an action is already loaded, a confirmation is requested backup.

**Delete Action …**
Displays the list of actions existantss, and can select what to delete. If the current action is deleted, the interface is reset with a new blank action.

**Save Action … / Save Action as …**
Displays the list of actions existantss, and can select what to delete. If the current action is deleted, the interface is reset with a new blank action.

**Back to home Page.**
Closes the editor and returns to action script.

### 9.2.2 Menu Preference



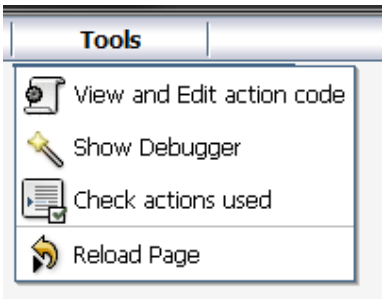

**Copy / Paste**
Copy / paste the selected command.
**Delete**
Deletes the command after confirmation.

**Line type …**
Defines the type of traits used to link these commands.

**Zoom x2 x1 x0.5…**
Allows you to zoom in or zoom out the workspace.

**Expand / Reduce Height**
To increase (expand) or decrease (reduce) the vertical work area.
**Expand / Reduce Width**
To increase (expand) or decrease (reduce) horizontally to the work area.
***Note: scrollbars only appear when the mouse is on the right side or on the bottom of the window***.

### 9.2.3 Menu Tools




**View and Edit action code**
Used to display the XML code of the current action. The XML code is created based on orders placed over the work area.
***Note: This feature is aimed at expert users. Incorrect changes to the XML can make the work unreadable***.
**Show Debugger**
Can display the Debug window. See below.
**Check Actions used**
Can view the display window of the use of actions.
**Reload page…**
Allows you to refresh the page.

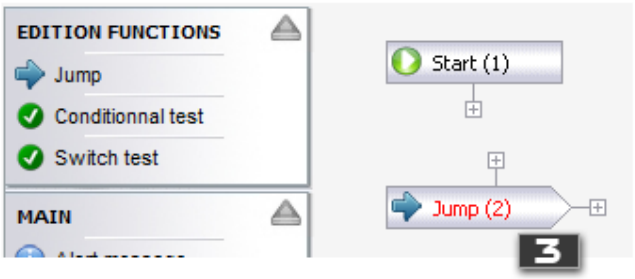
## 9.3 Use of the interface

By default, when creating a new action, only one command is displayed. This command: 'Start', defines the entry in the action, and contains no parameter.

### 9.3.1 Ajout de commandes

**1 –** Select the order in which you want to add a new command (in our example: Jump), it appears in red.

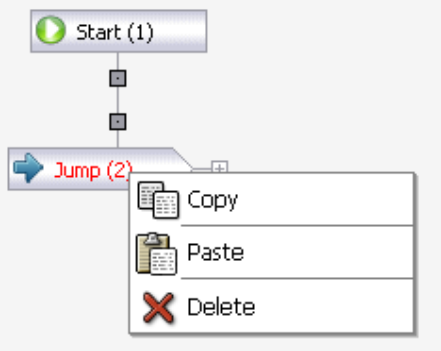
**2 –** Click the command you want to add.




**3 –** The command appears on the workspace, under the previously selected.

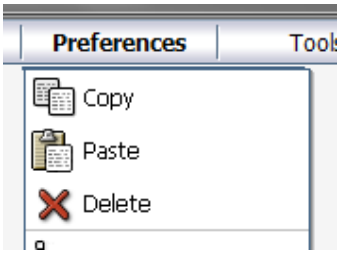
**4 –** Moved the new order (keeping the mouse button) until the two connections are displayed in green. Then release the click of the mouse, the two orders is automatically positioned.

### 9.3.2 Suppression de commandes.




**The selected command can be deleted:**

Either by a right mouse click on the command, then 'Delete'.



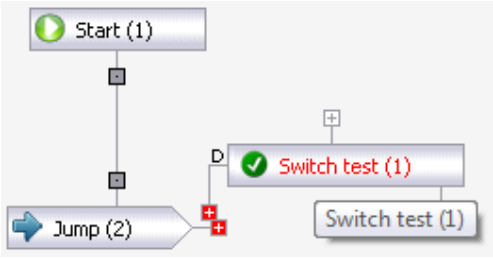

Either through the Edit menu> 'Delete'

### 9.3.3 Removing links between commands.

To delete a link between command simply double-click on one of two connections.

### 9.3.4 Illegal connections.

When two orders can not be connected together, each of the connections becomes red. Example :




Trying to connect two outputs commands them.

### 9.3.5 Various edition commands.

Actions panel orders
**Move the panel**: Using the drag and drop on the section headings (Edit, Main, DataBase, etc.).
**Replace the panel** in its original location (top left): Double click on the title of the section to reposition the panel top left.
**Open or close** different sections: Click on the blue arrow located next to the name of the section.

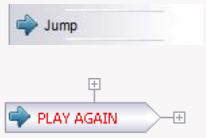
Actions orders
**Enlarge / shrink an icon**: Move the mouse over the command to resize, then use the mouse wheel while pressing the button 'Shift' key.
**Rotate icon**: Move the mouse over the command to rotate, then use the mouse wheel while pressing 'Alt' key.

## 9.4 Les commandes

### 9.4.1 Commands for editing and testing.

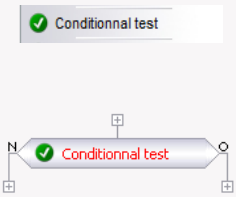
Remarks
The backup settings for a command is carried out:
1 - loss of focus on an input field.
2 - Pressing the key 'Enter' key in the input field.
3 - click on the 'floppy' panel setting.
When an order is changed and not saved, the text that displays the type of control is shown in red.

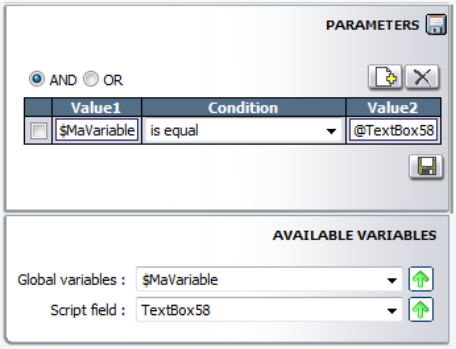
**A. JUMP**



**Description:**
This command allows you to continue the command to another script in progress.
**Parameter (s):**
The dropdown parameter panel contains the commands currently available in the script.

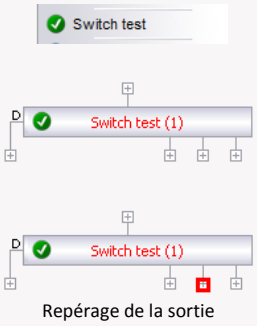
**B. CONDITIONNAL TEST**

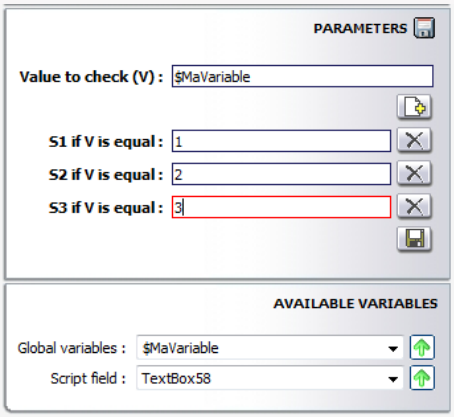



**Description:**
This command allows you to test a variable and redirect the script to a command based on the outcome of the test. If it's true, we use the output 'O' or exit 'N'.
**Parameter (s):**
To enter the function's parameters using the criteria editor Hemes. Net. For more information about this publisher see Annex 13.

### C. SWITCH TEST

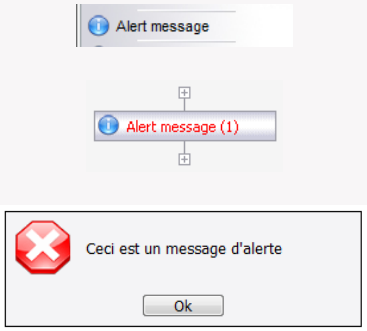


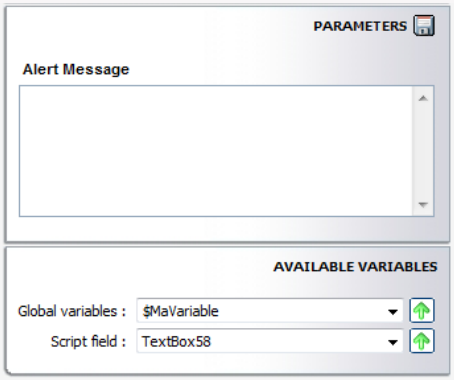


PARAMETERS
Value to check (V) : $MaVariable
S1 if V is equal : 1
S2 if V is equal : 2
S3 if V is equal : 3
AVAILABLE VARIABLES
Global variables : $MaVariable
Script field : TextBox58

**Description:**
This command allows you to test a variable and redirect the script to a command according to a number of possible results. Click on the 'More' to add new releases. The script is headed to the exit marked 'D' if the variable tested did not match any value entered for S1 to Sx.
**Note:** To find out what is the test, click the input field, the output is displayed in red. Fig. identification of the exit.
**Parameter (s):**
The parameters to enter are: the variable to be tested, and values corresponding to different outputs.

## 9.4.2 Main action commands.

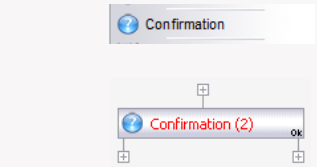
### D. ALERT MESSAGE

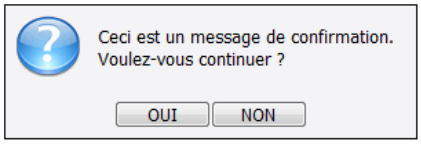


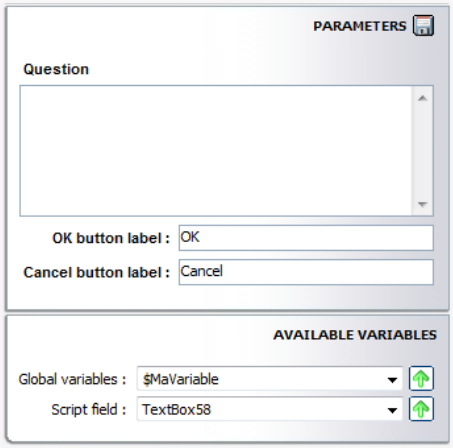




**Description:**
This command displays a message on the screen. The window display has an 'OK' button which allows you to delete it.
**Parameter (s):**
The text of the message or the variable display.

## E. CONFIRMATION

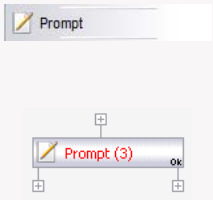


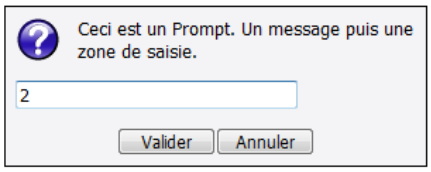



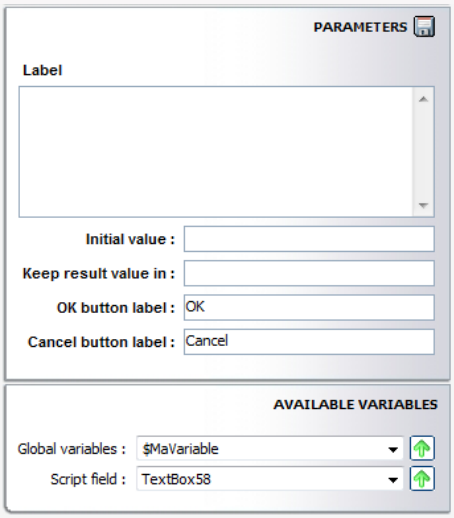




**Description:**
This command displays a confirmation message on the screen.
**Parameter (s):**
The text of the message or the variable display, the wording of the 'OK' button and the wording of the button 'ABANDONMENT'.

## F. PROMPT

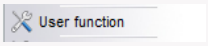


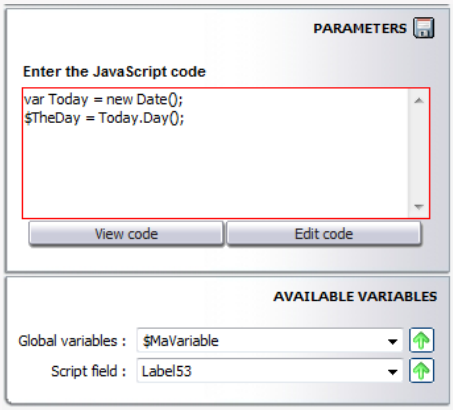




**Description:**
This command displays a message and an input box on the screen.
**Parameter (s):**
The text of the message or the variable display, the wording of the 'OK' button and the wording of the button 'ABANDONMENT'.

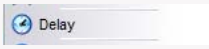
## G. USER FUNCTION

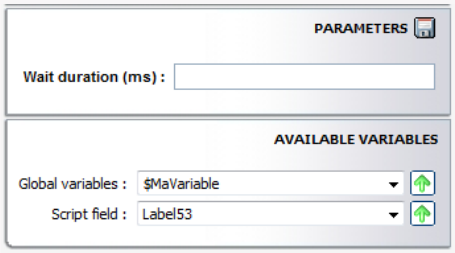





**Description:**
This command allows you to enter custom JavaScript code.
**Parameter (s):**
The JavaScript code.
The button 'View Code' allows you to display a window containing your code colorized version.
The button 'Edit Code' allows you to edit your code in a larger window and therefore more practical.

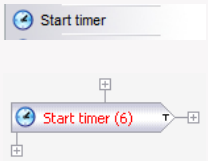
## H. DELAY

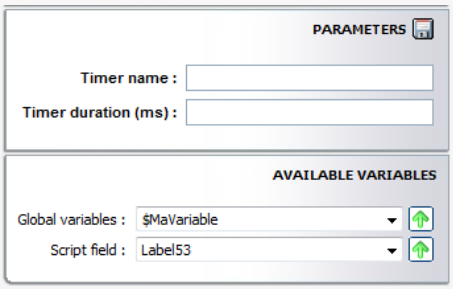





**Description:**
This command lets you define a period during which the action stops.
**Parameter (s):**
The duration in milliseconds of the stop.

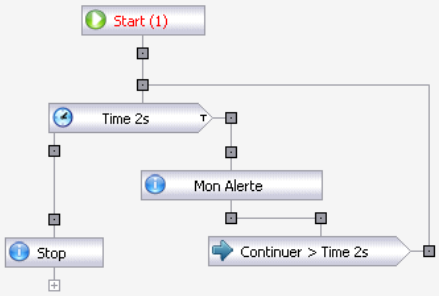
## I. START TIMER



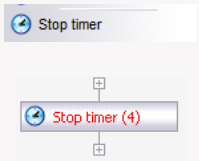



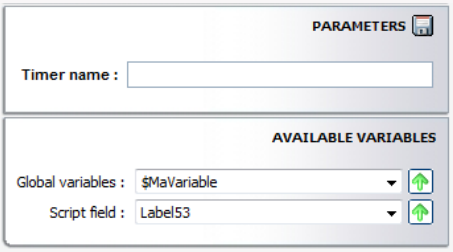

**Description:**
This command allows you to create a loop, repeated at regular intervals.
**Parameter (s):**
Give a name for the timer loop. Indeed, this name will be used to stop the loop.
Millisecond interval repetition of the loop.

When the timer is stopped by a 'Stop Timer' action continues through the left exit of the order.




## J. STOP TIMER

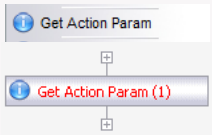


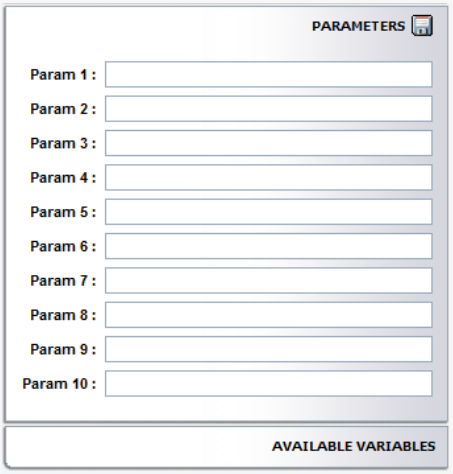




**Description:**
This command stops a timer.
**Parameter (s):**
The name of the timer specified in the 'Start Timer'.

### 9.4.3 Commandes avancées.

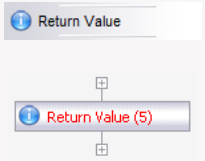
#### K. GET ACTION PARAM

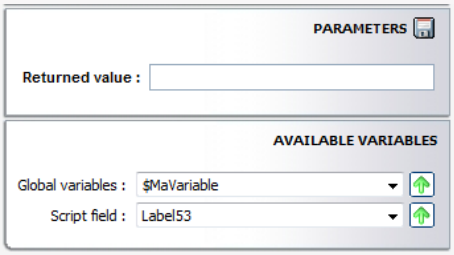



**Description:**
This command allows you to assign local variables to the parameters passed to the action if it is called from another action. (see command 'Execute Action').
**Parameter:**
The 10 parameters max. entry in the action.

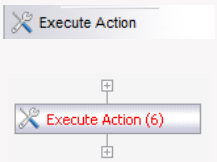
#### L. RETURN VALUE

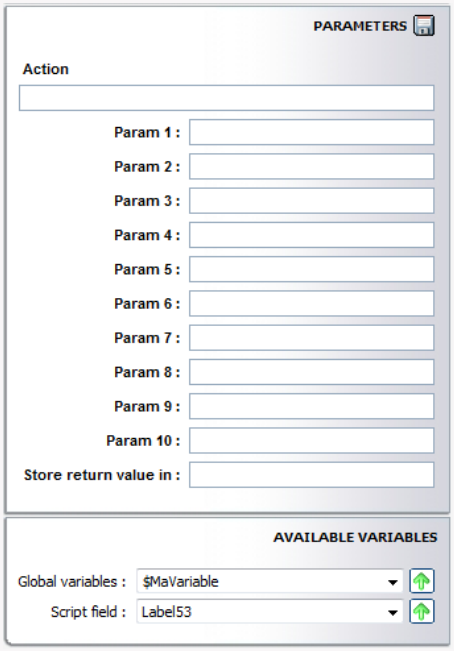



**Description:**
This command lets you specify a return value of the action. (see command 'Execute Action').
**Parameter:**
The value to return.

## M. EXECUTE ACTION



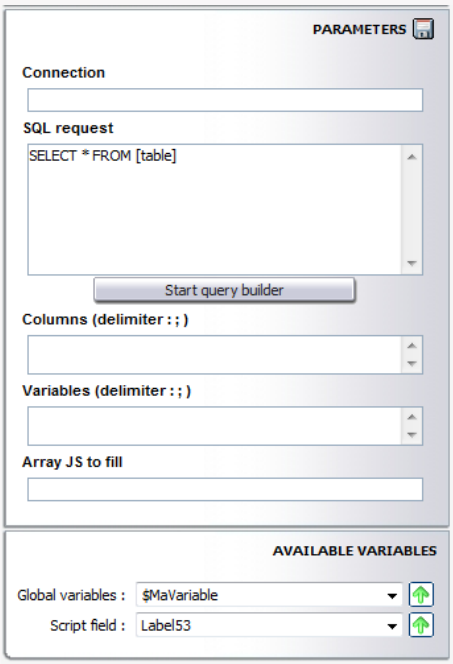




**Description:**
Allows you to call another action in passing up to 10 parameters. If the action returns a known value, it is possible to store this value in a global variable.

**Parameter (s):**
The name of the action to call.
The 10 parameters (optional)
Global variable used to save the setting back (optional).

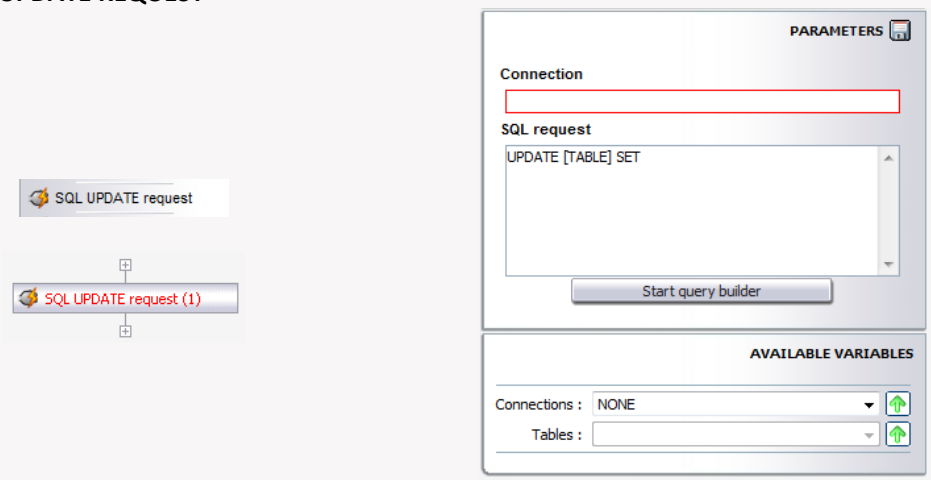
### 9.4.4 Database commands.

**N. SQL SELECT REQUEST**




**Description:**
Executes a select request on a table. The columns specified in the request are assigned to global variables. The full results of the request may be used to populate a JavaScript array.
**Parameter (s):**
- The connection to use. To do this use the drop-down list of the variables to select one of the connections available.
- The query to execute. You can help the "query builder" described in Annex 8.
- The list of columns returned by the query and you want to assign to global variables.
- Global variables to affect (in the order of columns defined above)
- The global variable that contains the complete table for the query.

## O. SQL UPDATE REQUEST

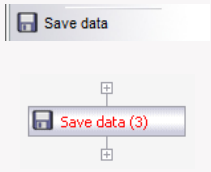



**Description:**
Executes a request to update on a table.
**Parameter (s):**
- The connection to use. To do this use the drop-down list of the variables to select one of the connections available.
- The query to execute. You can help the "query builder" described in Annex 8.

## P. SQL INSERT REQUEST

**Description:**
Executes a request for insert on a table.
**Parameter (s):**
- The connection to use. To do this use the drop-down list of the variables to select one of the connections available.
- The query to execute. You can help the "query builder" described in Annex 8.

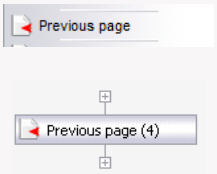
## Q. SAVE DATA

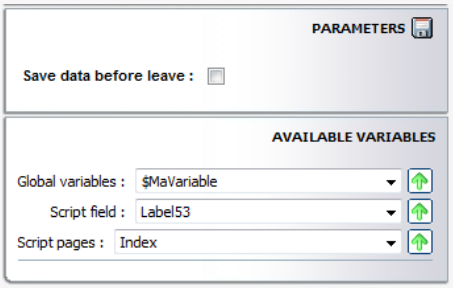



**Description:**
Written data page in the database. Equivalent to a navigation command, all fields related to a table are saved.
**Parameter (s):**
None

### 9.4.5 Navigation commands.

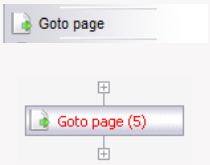
**R. PREVIOUS PAGE**

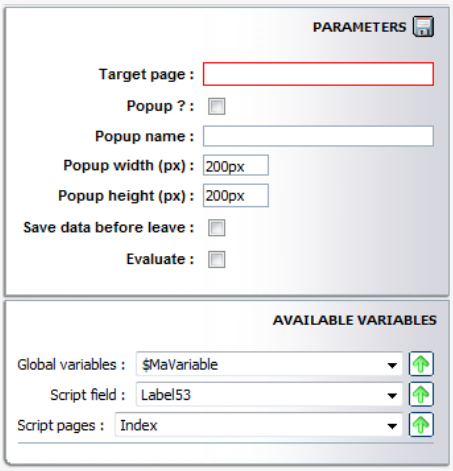



**Description:**
Allows you to navigate to the previous page.
**Parameter (s):**
Check this box if you want to back up the page before returning to the previous page.

**S. GOTO PAGE**



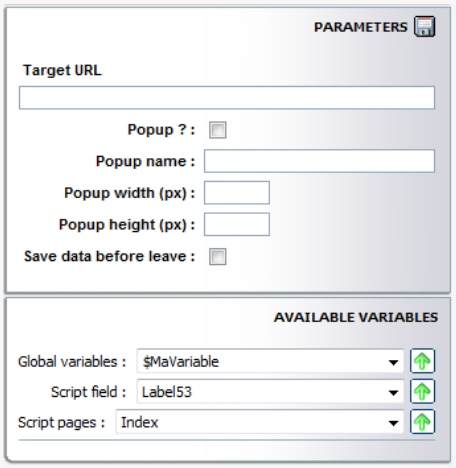




**Description:**
Allows you to navigate to another page of the script.
**Parameter (s):**
- The destination page. It must be selected in the drop-down list displayed in the 'Variables'.
- Check the 'Pop-up' if you want the navigation is done in a new browser window, type pop-up.

**Where the Pop-up option is selected:**
- The name of a pop-up can be specified. This name is useful if you want to enable or disable the focus to this window.
- The width and height of the window can be specified.
- Check the 'Save data ..' to save the data on the page before the navigation.

- Check the 'Evaluate' in order to enter a variable in the field 'Target Page'.

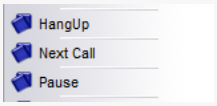
## T. GOTO URL

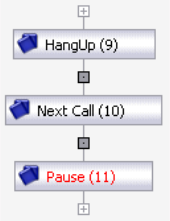



**Description:**
Allows you to navigate to an Internet address free.
**Parameter (s):**
- Check the 'Pop-up' if you want the navigation is done in a new browser window, type pop-up.
**Where the Pop-up option is selected:**
- The name of a pop-up can be specified. This name is useful if you want to enable or disable the focus to this window.
- The width and height of the window can be specified.
- Check the 'Save data ..' to save the data on the page before the navigation.

### 9.4.6 CTI Commands .

#### U. HANGUP / NEXT CALL / PAUSE

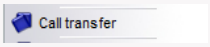


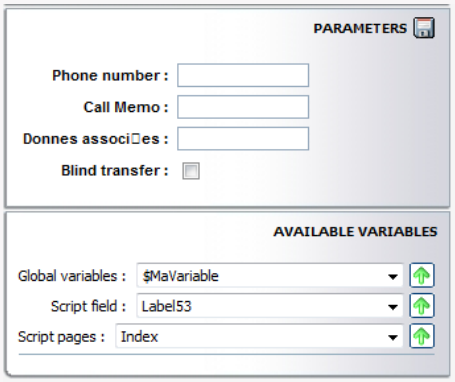




**Description:**
Identical to the actions of the bar phone.

No parameters required.

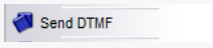
#### V. CALL TRANSFERT

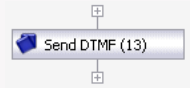


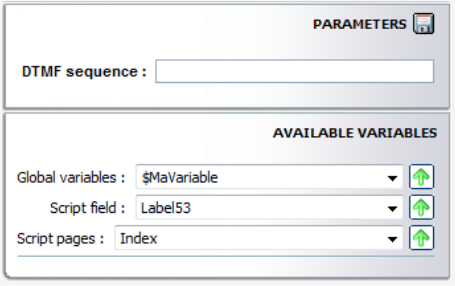



**Description:**
Can transfer the call to an external number, an SDA, an agent.
**Parameter (s):**
- The number of: such, SDA, or an agent.
- CallMemo and related data can pass data to the recipient.
- Do not check the box 'Blind Transfer' to perform a search.

#### W. CALL TRANSFERT

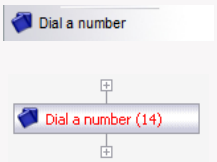



**Description:**
Can generate a DTMF code during the call.
**Parameter (s):**
The DTMF sequence to play.

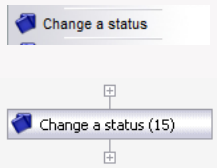
### X. DIAL A NUMBER

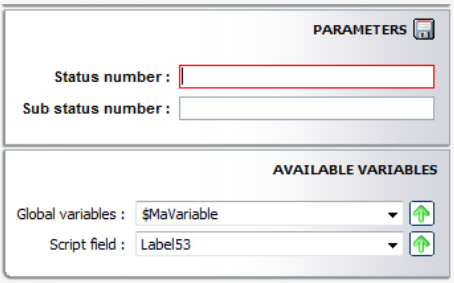


**Description:**
Allows to call a phone number. (External, SDA, Agent).
**Parameter (s):**
Number to call.

## 9.4.7 Advanced CTI Commands .

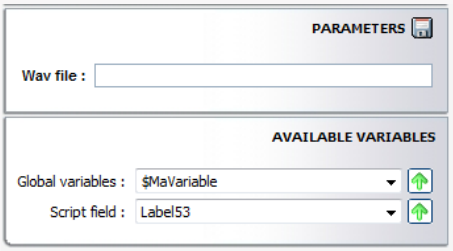
### Y. CHANGE A STATUS



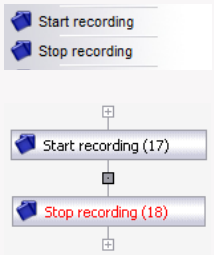

**Description:**
Allows you to decide the call. To know the status numbers, see the administration interface. The fact of the appeal decision in the script and before the end of the call, providing you have it in after the call.
**Parameter (s):**
Number of status and sub-status.

### Z. PLAY WAV



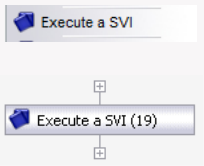

**Description:**
To play a wav file present on the ACD.
**Parameter (s):**
Full name of wav file.

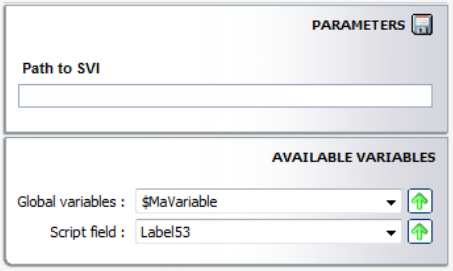
## AA. START / STOP RECORD




**Description:**
Starts and stops recording of the conversation. The names of files for registration are those defined in the administration.
**Parameter (s):**
None

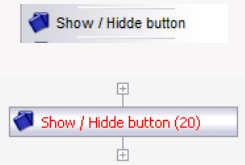
## BB. Execute an IVR



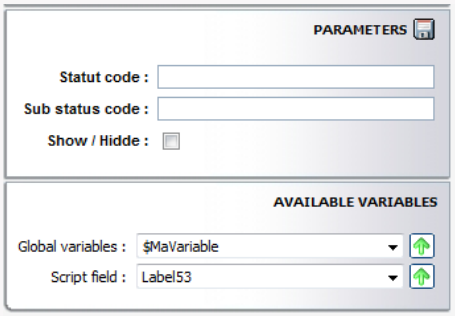

**Description:**
Run a IVR during the call.
**Parameter (s):**
Full path and name of IVR.

## CC. SHOW/HIDDE BUTTON



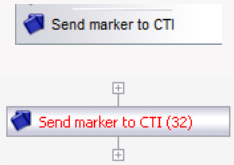
**Description:**
To show or hide the buttons this phone.
**Parameter (s):**
Button code and state of the button.

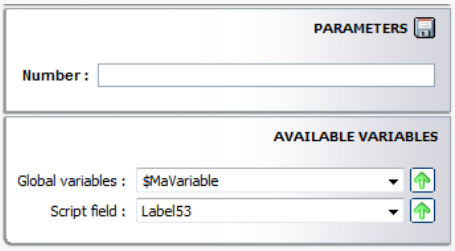
## DD. SHOW/HIDDE STATUT




**Description:**
To show or hide the status of qualification. Very practical to filter the list of statutes based on the location in the script.
**Parameter (s):**
Status code, sub-status, state status.

## EE. SEND MARKER TO CTI

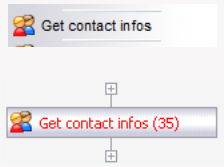


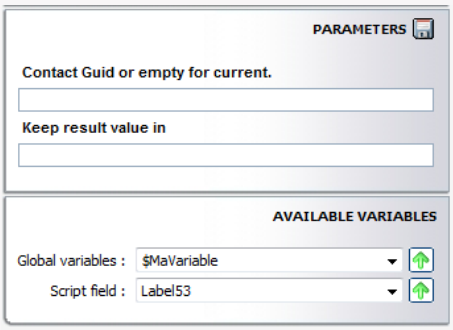




**Description:**
Allows you to send a marker to the real time supervision. This marker passes through the proxy then CT is received as an agent through supervision that can display it.
**Parameter (s):**
Number is the text to display in the screen of supervision.

### 9.4.8 CRM Contact Commands.

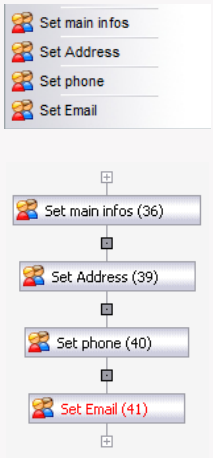
**FF. GET CONTACT INFOS**

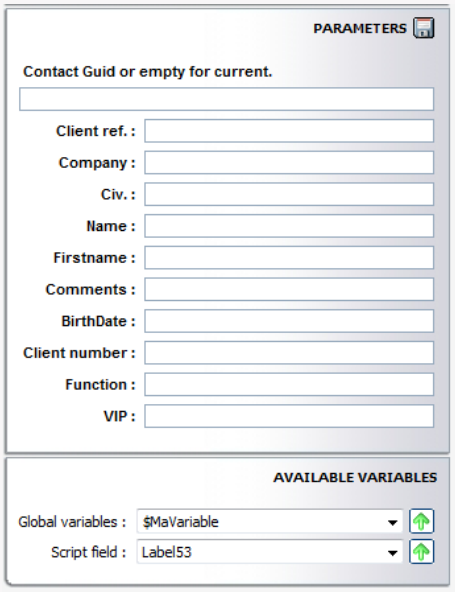



**Description:**
Allows you to recover the information about the online contact or another contact whose GUID is known. The Contact object is defined below and is placed in the selected variable.
**Parameter (s):**
Contact guid or empty string for the current contact.
The variable that contains the contact information.
See Structure of the Contact object.

**GG. SET MAIN INFOS / ADDRESSES / PHONE / EMAIL**




**Description:**
Can update an object on the contact line or contact another contact whose GUID is known. The Contact object is defined below and is placed in the selected variable.
**Parameter (s):**
Contact guid or empty string for the current contact.
The parameters to be updated according to the order.
See Structure of the Contact object.

### 9.4.9 Information on command CRM Contact.

The Contact object returned by the CRM commands has the followinw structure :

```
function Consumers() {
   this.ReferenceClient = "00000000-0000-0000-0000-000000000000" / / guid Contact
   this.Company = "";                          / / Company Contact
   this.Civility = "";                         / / Civilitée Contact
   this.Name = "";                             / / Contact Name
   this.FirstName = "";                        / / First Name
   this.Comment = "";                          / / Comment in contact
   this.BornDate = "",;                        / / Date of Birth of contact (free format)
   this.Adresses = new Array ();               / / Table containing the addresses of the contact
   this.Tels = new Array ();                   / / Table containing phones Contact
   this.Mails = new Array ();                  / / Table containing the contact's e-mail
   this.Cookies = new Array ();                / / Table containing cookies internet contact
   this.ClientNumber = "";                     / / Number of the client (free format)
   this.Job = "";                              / / Post Contact
   this.IsVIP = 0;                             / / Contact VIP (0: false 1: true)

}
```

Member "Addresses" is a table containing the addresses of the contact. Each address is an instance of the class javascript "Addresses" defined as follows:

```
function Adresses() {
   this.AdressStr = "";    //  address, may be on several lines (line separator: \ n)
   this.ZipCode = "";      //  Zip Code (free format)
   this.City = "";         //  Ville de l'adresse
   this.State = "";        //  State of the address
   this.Country = "";      //  Country of the address
   this.Type = "";         //  Type of address for the contact (example: "Home").
   }
```

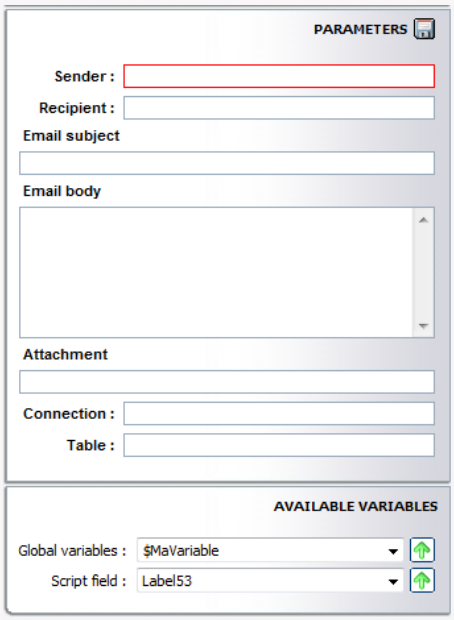
Member "as" is a table containing the contact phone number. Each phone is an instance of the class javascript "ContactMedias" defined as follows:

```
function ContactMedias() {
    this.TelType = "";      //  Type the phone for the contact (example: "Home")
    this.TelNumber = "";  //  Phone Number
   }
```

Member "mail" is a table containing the contact e-mail. In this case, the "TelNumber" will contain an email address.
Member "Cookies" is a table with cookies of the contact. In this case, the "TelNumber" will contain the value of the cookie.

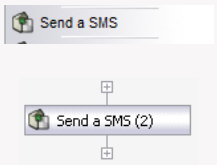
### 9.4.10 V.M.C Commands .

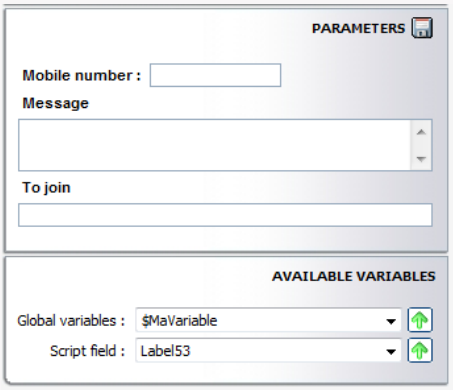
**HH. SEND A EMAIL**




**Description:**
Allows you to send an email from the script.
**Parameter (s):**
The address of the sender (Sender), recipient (Recipient).
The subject of the mail.
The body of the mail. It is possible to use the marker @ @ NOM_DU_CHAMP to merge the body of the message with the contents of the table provided below.
A / attachments (Full path of the attachment)
Connection Table, to indicate the information to merge if there are a. The key used for the query is the index of the record of the script in progress.

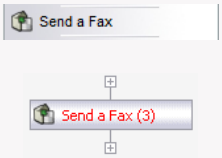
## II. SEND AN SMS

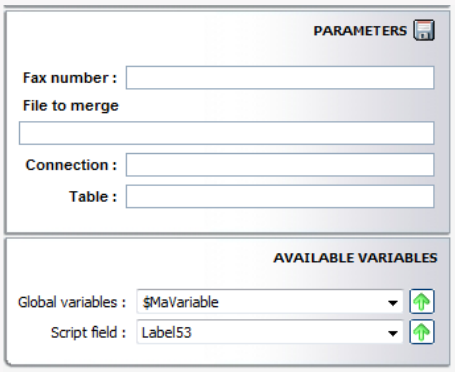



**Description:**
Can send an SMS from the script.
**Parameter (s):**
The number of mobile phone.
The message.
The attachment for MMS

## JJ. SEND A FAX



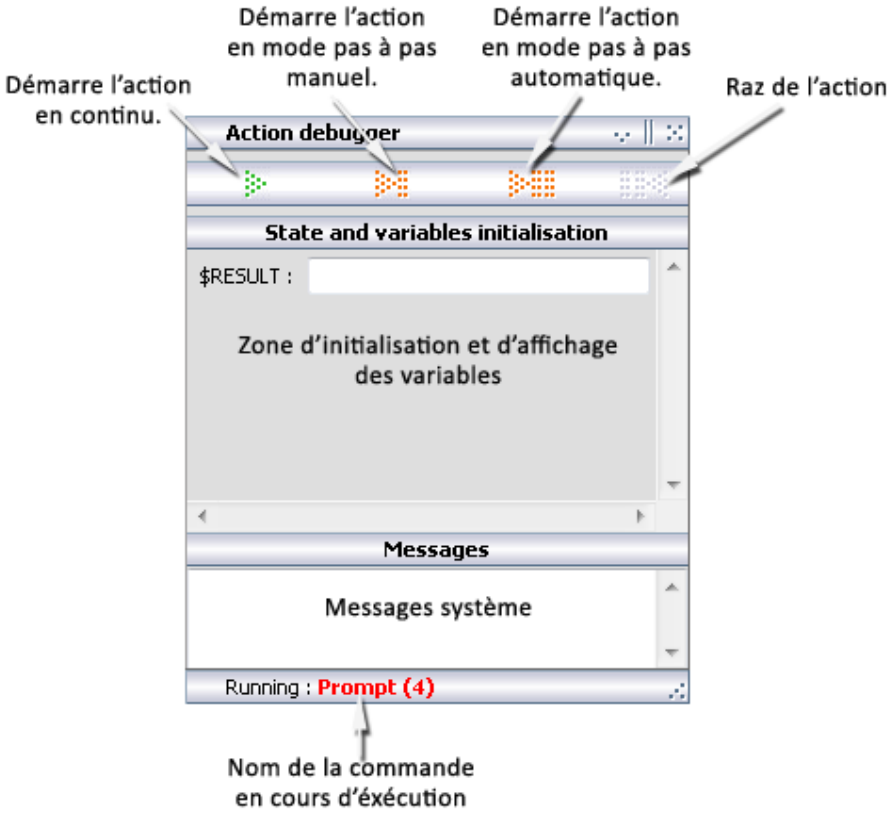

**Description:**
Allows you to send a fax from the script.
**Parameter (s):**
The fax number of the customer.
The file to merge in the fax.
It is possible to use the marker @ @ NOM_DU_CHAMP to merge the file with the contents of the table provided below.

## 9.5 action Debugger

### 9.5.1 Principle

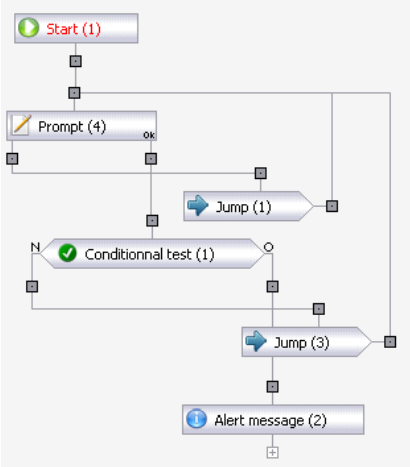
The debugger allows action to follow step by step the progress of an action. You can initialize the state of global variables and field values of the script if the action involved.

### 9.5.2 Interface


Démarre l'action en continu.
Démarre l'action en mode pas à pas manuel.
Démarre l'action en mode pas à pas automatique.
Raz de l'action
Action debugger
State and variables initialisation
$RESULT :
Zone d'initialisation et d'affichage des variables
Messages
Messages système
Running : Prompt (4)
Nom de la commande en cours d'éxécution

### 9.5.3 Example debug action

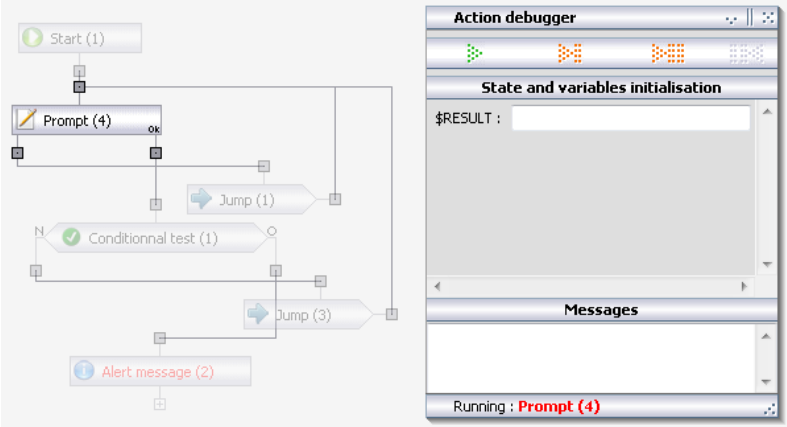
**the following action:**




Command Prompt
Output OK> Test the input value ($ result).
Cancel Exit> Back to the prompt.

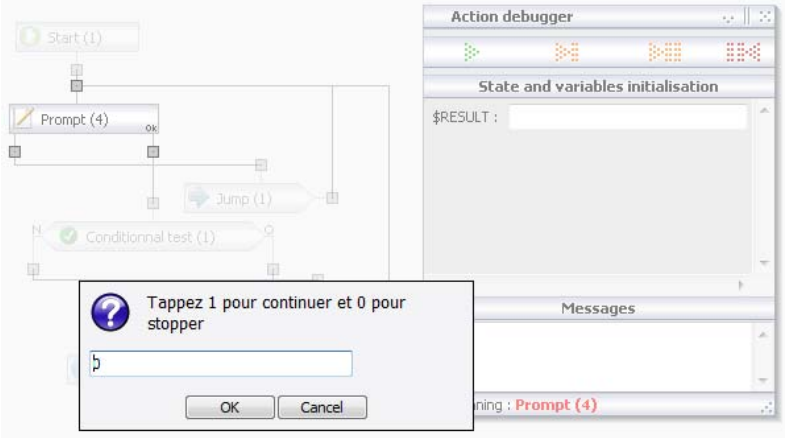
Conditional Test
$ Result = 1> Command Alert.
Otherwise> Back at the prompt.

**Action debug display.**
**A - Init**



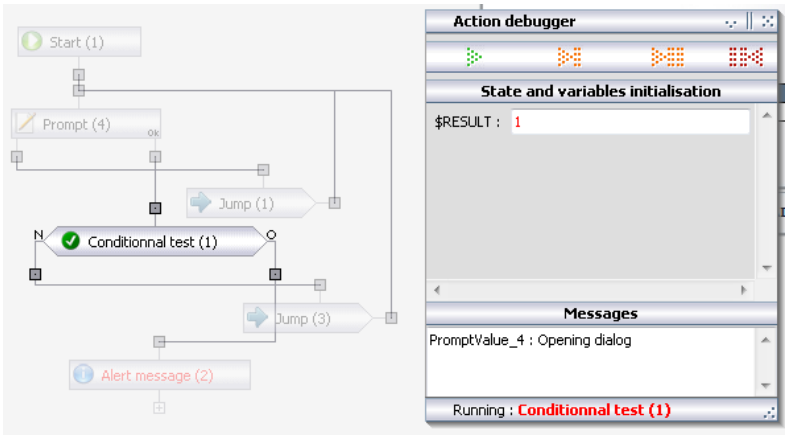

*At this stage it is possible to initialize the variables.*

**B - Launch Step by Step**



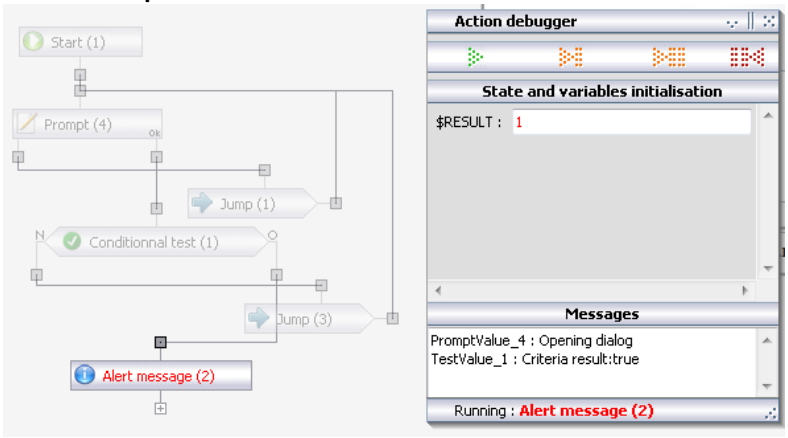
*The first command is executed.*

### C – Enter the value 1 and click "OK"



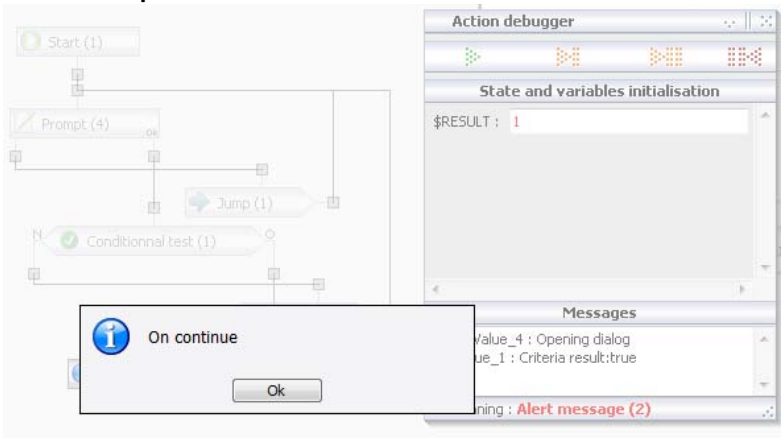

*We note that the variable $ result has taken the value entered. The action is to test the global variable. The debugger waits for a click on 'Next Step'.*

### D – Next Step



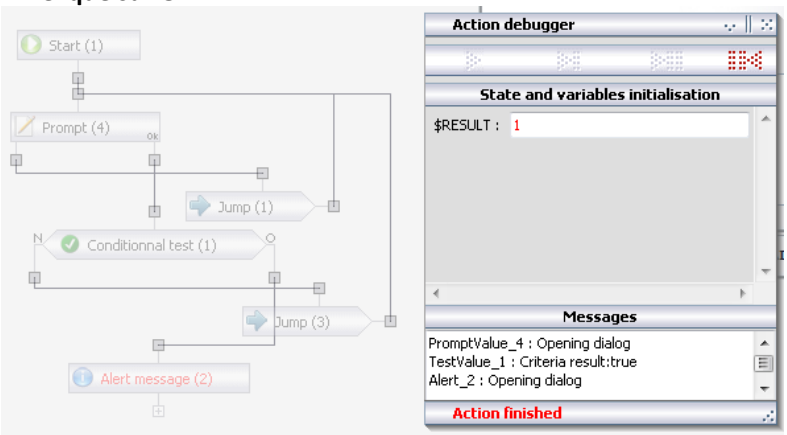

*The test is positive, the action continues on the outlet 'O' and is ready for the next command.*

### E – Next Step




*The 'Alert' is executed. The alert message appears.*

**F – Clique sur OK**



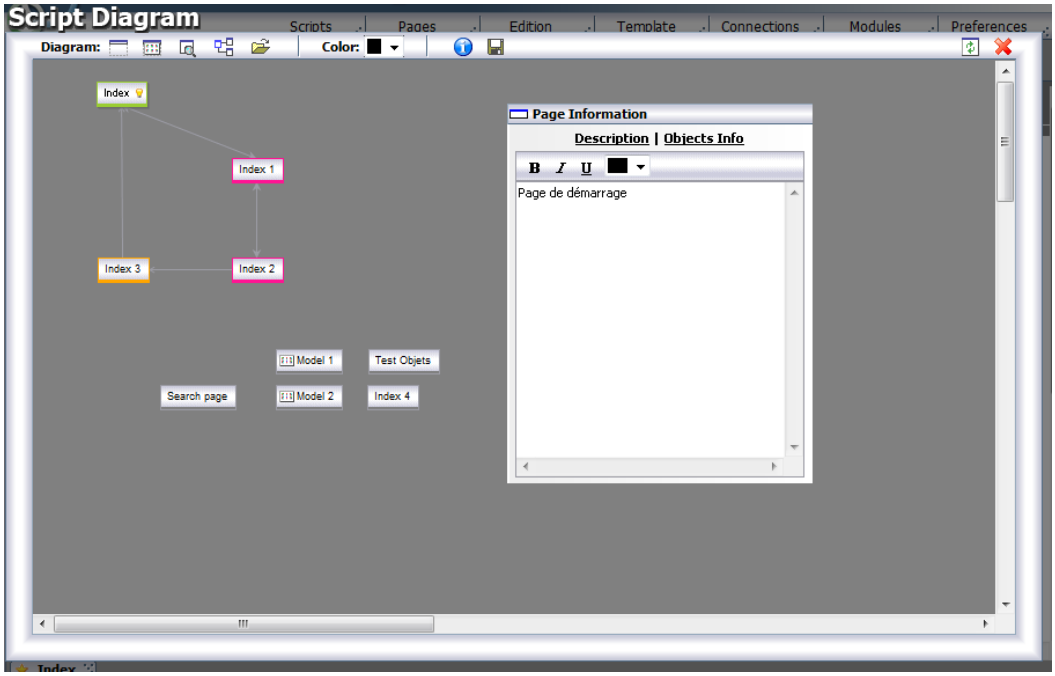

*The action is finished.*

# 10. Appendix « Diagram Viewer »

## 10.1 Description

Diagram viewer module allows you to have a comprehensive representation of the pages of the script.

It allows you to:
- To position the pages on the screen manually or automatically.
- Commented to each page in the description of the pop-up information.
- Set color page and giving the group a clearer picture of the script.
- To hide the template pages or unwanted in the diagram.
- To safeguard the performance.
- To know the information link between pages via the Object Info pop-up information.
- Print out the list of aggregated links per page with their destinations.
- Display screen superimposed relations overview page while the mouse icon.

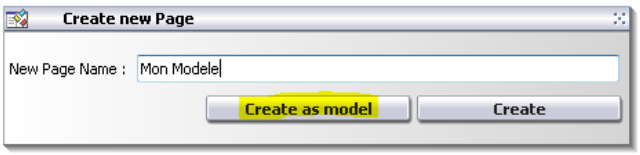


# 11. Appendix « Page template »

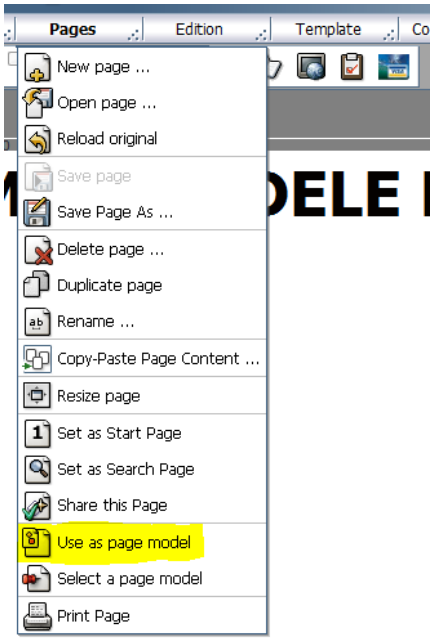
## 11.1 Création

The template pages of the interface designer Hermes. Net can create pages that will never be displayed on the screen when running the script but will serve as a basis or model for other active pages.

There are two ways to create a page template. The first is to click on the button 'Create as Model' in the window to create a new page:




The second way to proceed is to use an existing page and assign it the status of 'model':




At any time a page template can be reassigned as a standard page by unchecking the menu item.

## 11.2 Affectation

To assign a template to a standard page, select the template page in the pop-up Select a top model.
By clicking on the menu item 'Pages', pop-up appears.

This window displays the list of template pages are in the script.

Select the desired page and click 'Apply'.
Check the checkbox 'Apply to all model page' or automatically for all pages using the model chosen.

Click on the button 'remove' to remove the model of the current page.

The checkbox in the status bar to hide or display the model.

# 12. Appendix « Search mode »

The search script should allow at least retrieve the index of the client file to call, and telephone number.
The display of the search may be done with the subject "Data Grid".

To switch the call in the search mode, you must create an action type "User" category "Custom Functions".

The first step is to recover the object AgentLink using the function GetAgentLink.

**Usefull AgentLink methods:**

- SearchModeSelect(int indice, bool makeCall, object Number, object Memo, Object Data)
  - Indice : indice if teh number to call
  - makeCall : make call, or mount the script page in the "preview"
  - Number : phone number to call
  - Memo : Data Mémo to pass as a parameter in the outbound script
  - Data : Data to pass as a parameter in the outbouns script

- SetSearchMode(bool enable)
  - Enable : enables / disables the search mode

**Sample code to make a call from the search mode:**

```
var toolbar = GetAgentLink();  //get the activeX

if (toolbar != null)
{
        var att = @DataSearch.line; // get the selected row in the DataGrid
        $INDICE= att[0]; //get the indice  : first column of the DataGrid
        $TEL=att[4]; //get the phone number : fifth column of the DataGrid

        if (att[0] && att[4]) // Validity Test
        {
                toolbar.SearchModeSelect($INDICE, true, $TEL); //Calls the method
                SearchModeSelect to make //the call
        }
}
```

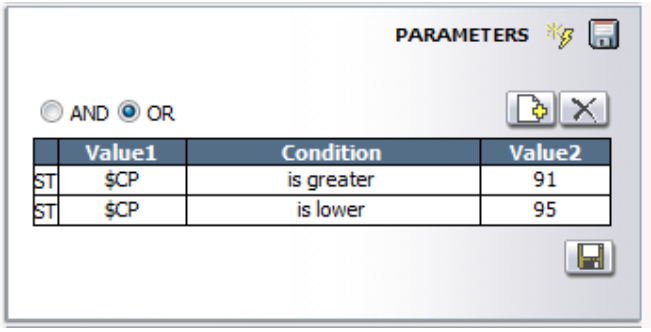
**Code Sample to close the search mode from the script:**

```
var toolbar = GetAgentLink(); // get the activeX

if (toolbar != null)
{
        toolbar.SetSearchMode(false); //disables the search mode
}
```

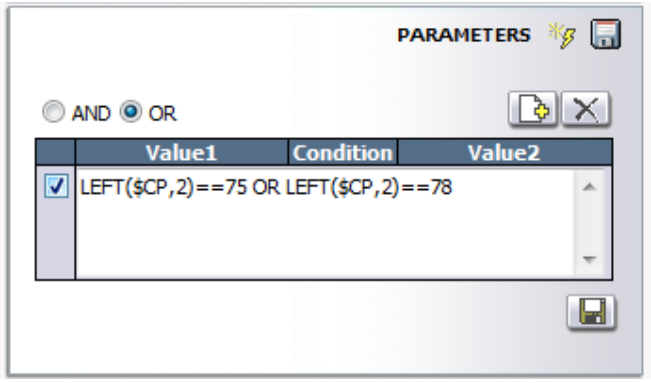
# 13. Appendix « criteria editor »

The editor of criteria used to generate tests on the variables.




Column 'value 1' contains the value to be tested, the column 'status to perform the test on this variable, and the column' value 2 'the value to test.
The radio button allows you to specify whether an 'OR' or 'AND' must be made between criteria.

By clicking the check box to the left of the table it is possible to write its own criteria, as follows:



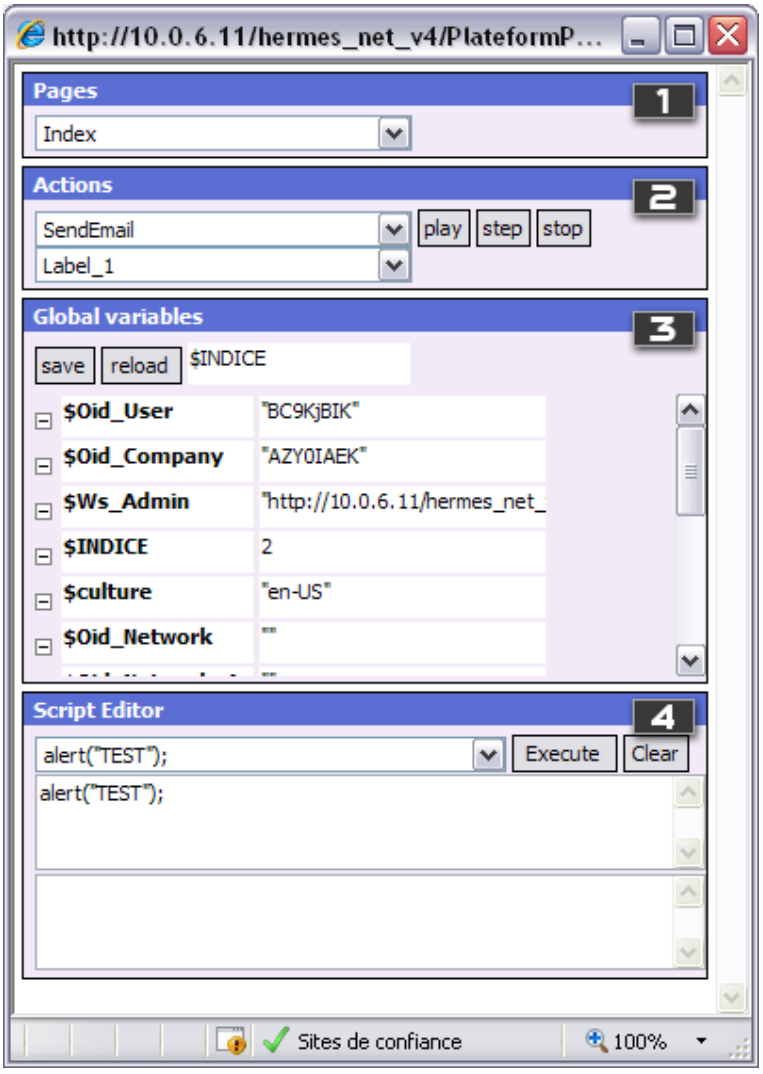
# 14. Appendix « Script Debugger»

The script debugger allows the testing of the generated script and know the values of global variables at different stages of the script.

Enabling the debugger is in the window generation script by checking the "mode control".

A picture to launch the debugger and then added to the script in the upper left ( ).Simply click on this image will open the debugger.

The debugger opens in a new browser window:




1. This list displays the list of pages in the script. By selecting a page in the list, the script navigates automatically to it.
2. This area displays all the actions of the script (first list) and for each share commands that compose the (second list).
The button "play" allows to perform the selected action: all orders are executed.

The "Step" button allows you to run the selected command only. At the end of the stage you can see the value of global variables and see how the order could change.
The button 'stop' to stop the execution of the order (if an execution stage has been selected).
3. This area will display all the global variables of the script as well as their value.
You can change the value of variables in the text field in front of the variable.
If the variable is changed, the color of the text field is changed to blue.
The button "save" is used to validate the modified values of global variables.
The button "reload" allows you to cancel the manual modification of the values of global variables.
4. This area allows the user to enter the free javascript code.
The selection list lets you select a command already typed.
The first text field lets you enter the javascript code to execute.
The second text field displays the result of the function (if the code executed returns a value).


Vocalcom - 7 rue de Tilsitt – 75017 Paris - Tél. : +33 1 55 37 30 50 – Fax : +33 1 55 37 30 51 – www.vocalcom.com
SA au capital de 5 000 000 € – RCS Paris B 401 973 631 – TVA Intracommunautaire : FR 27401973631

# 15. Aappendix « create your own style sheet »

## 15.1 Import CSS files

To allow the script to use your style, you must first send to the server. For this, use the module "Download files" available from the menu "Modules".
A section specific stylesheet is present in this module.

Once uploaded the file, you must specify InterfaceDesigner that this file must be applied. To do this go into the menu "Scripts" and select "style script. In the popup, select the style file you upload.

If, after uploading the css file you wish to change it, you have to do locally on your computer, then upload it again. The upload module allows you to file to re-enter if necessary.
Once you've uploaded the file, you must select it again in the list of styles if you want to see the result directly in InterfaceDesigner. In all cases, the file will be changed when generating or after reopening the script in InterfaceDesigner.

## 15.2 Select a style to apply to script

By default, all objects are created the script with a class style called "Nom_Type_Objet_En" _HN. Thus, in your style changed, if you want to combine objects with the default style and the new, you must keep the default styles of the sample file (attached with this document) and add to your own classes style.

Each object in the property "class style" in the property "Appearance." You can choose another style class, corresponding to a class that you created.

The default styles classes are:

- Object **Label**: Label_HN
- Object **panel** : Panel_HN
- Object **TextBox**: TextBox_HN
- Object **Date**: Date_HN
- Object **Time**: Time_HN
- Object **Textarea**: Textarea_HN
- Object **Combo**: Combo_HN
- Object **Listbox**: Listbox_HN
- Object **free zone** : non applicable
- Object **button** : Button_HN
- Object **Radio**: Radio_HN
- Object **Checkbox**: Checkbox_HN
- Object **hyperlien** : Link_HN
- Object **image** : pas de classe de style par défaut
- Object **Datagrid**: Datagrid_HN
- Object **objection** : Objection_HN
- Object **IFrame** : pas de classe de style par défaut

- Objet **agenda** : non applicable
- Objet **credit card** : non applicable

## 15.3 Object structure in DOM and names of the classes used

You can change the class of default style for this object. For our examples we will call the custom class' ClassPerso.

### 15.3.1 Object « label »

The object "label" of InterfaceDesigner corresponds to the object HTML "span". It is composed of that element only.
You can define a custom class as follows:

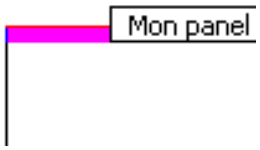
*span.**ClassPerso** {…*
*}*

### 15.3.2 Object « panel »

L'objet « panel » d'InterfaceDesigner est composé de plusieurs éléments.

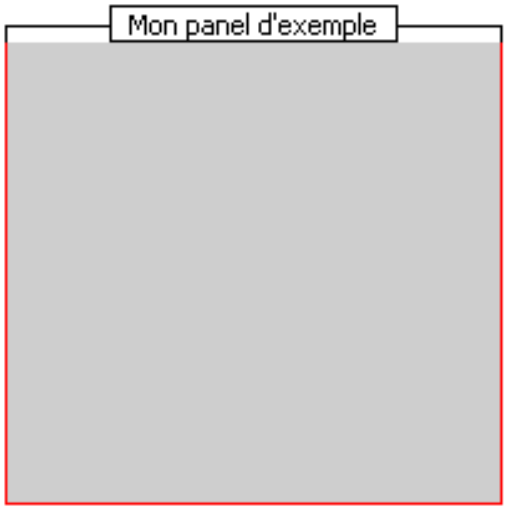
The part that displays the title of the panel corresponds to the HTML object "td". The class style used is the class: « td.**ClassPerso**_title ».

The upper left panel to display the "title" of the panel centered vertically, corresponds to the object HTML "div". The class uses the class « div.**ClassPerso**_top_left » . This area corresponds to the rose in the example below :




The upper right panel to display the "title" of the panel centered vertically, corresponds to the object HTML "div". The class used is the class « div.**ClassPerso**_top_right ». This area corresponds to the fushia in the example below :

The central part of the panel corresponds to the object HTML "div". The class uses the class "div.ClassPerso_bottom". This area corresponds to the gray edged with red in the example below:




It is also possible to define common properties in three types of "div" described above by defining a style for "table.ClassPerso div":
table.ClassPerso div*{*
*...*
*}*

### 15.3.3 Object « text field »

*The object "fields" of InterfaceDesigner corresponds to the object HTML 'input' is composed of that element.*
*You can define a custom class as follows:*
*input.ClassPerso{*
*...*
*}*

### 15.3.4 Objet « date field »

The object "date fields" is composed of several elements.

The object "date fields" is primarily composed of an object HTML "div" elements in its container. It is for example possible to apply a border by changing the general class "div.ClassPerso" such:

*div.**ClassPerso** {*
*padding:5px;*
*border:2px solid black;*
*}*

Both sides of the object (the text box and button) are contained in objects of type "td" (one for each different td). It is therefore possible to play on the spacing between the two parties or to apply a background color. For example:

*div.**ClassPerso** td {*
*padding-left:5px;*
*padding-right:5px;*

*background-color:green;*
*}*

Part input area corresponds to the HTML 'input'. You can customize it by defining a style for "input.ClassPerso_text".

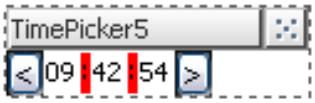
The button part is the HTML object "input". You can customize it by defining a style for "input.ClassPerso_button".

### 15.3.5 Object « Time picker »

The object "time picker" is composed of several elements, all contained in the HTML element "table" itself in an HTML element "div". It is therefore possible to define a border general subject or change the background color of any object by setting the style "div.ClassPerso table.

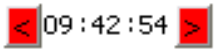
The various components of the object "field-hour" are contained in objects of type "td" (one for each different td). To change these containers, defining the class: "div.ClassPerso table td.

Separators hours are objects of type "div". You can change them by defining the class "div.ClassPerso_text", as in this example against:




Indicators of time (hour, minute and second) are objects of type "input". You can change them by defining the class "input.ClassPerso_text", as in this example against:

Finally, the two buttons to increase or decrease the time assets are of type "input". You can change them by defining the class "input.ClassPerso_button", as in this example against:




### 15.3.6 Objet « Text area »

The object "multiline text" of InterfaceDesigner corresponds to the object HTML "textarea". It is composed of that element.
You can define a custom class as follows:
textarea.ClassPerso*{*
*…*
*}*

### 15.3.7 Object « combo »

The object "combo" of InterfaceDesigner is composed of different objects. In the script generated, the object used is an HTML "select". To keep an identical behavior between the script and editing the script generated, it is appropriate to use the generic class in the css file:

.***ClassPerso*** *{*
*…*
*}*

### 15.3.8 Object « list box»

The object "multiselection list of InterfaceDesigner is composed of different objects. In the script generated, the object used is an HTML "select". To keep an identical behavior between the script and editing the script generated, it is appropriate to use the generic class in the css file:

*.**ClassPerso** {*
*...*
*}*

### 15.3.9 Objet « button »

The object "button" of InterfaceDesigner corresponds to the object HTML 'input'. It is composed of that element.
You can define a custom class as follows:
*input.**ClassPerso** {*
*...*
*}*

### 15.3.10 Object « radio button»

The object "button" of InterfaceDesigner consists of several components all HTML content in an element type "table". It is therefore possible to modify the general look of the object by modifying the class "table.ClassPerso.

The radios are viewing the HTML element "input". It is therefore possible to change their style by changing the style table.ClassPerso input.

The texts are displayed in the HTML element "label". It is therefore possible to change their style by changing the style table.ClassPerso label. "

### 15.3.11 Objet “Check box”

The object "check box" of InterfaceDesigner consists of several components all HTML content in an element type "table". Its operation is identical to the object "radio button" described above. It is therefore possible to modify the general look of the object by modifying the class "table.ClassPerso.
The radios are viewing the HTML element "input". It is therefore possible to change their style by changing the style table.ClassPerso input.
The texts are displayed in the HTML element "label". It is therefore possible to change their style by changing the style table.ClassPerso label. "

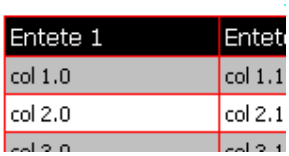
### 15.3.12 Objet « Hyperlink »

The object "hyperlink" to InterfaceDesigner corresponds to the HTML object "a". It is composed of that element.
You can define a custom class as follows :
*a.**ClassPerso** {*
*...*
*}*

### 15.3.13 Object « data grid »

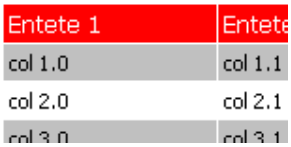
The object "data grid" is a complex composed of various elements. Furthermore it allows a definition of style more advanced than the HTML element "table" standard, such as the differentiation of styles between odd and even lines.

All the objects comprising the data table are grouped in the object HTML "div". It is therefore possible to change the background color or the border by changing the general style

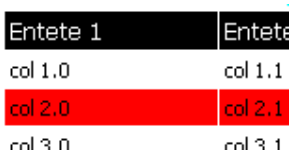



« div.**ClassPerso** », such as changing the background color:

You can customize the header row of the table data by modifying the style div.ClassPerso thead ", such as changing the background color:

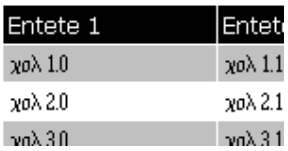



You can customize the style pair of lines (not counting the header as a first line) by changing the style tr.ClassPerso_Line_0 ", such as changing the background color:




The style of the odd lines (not counting the header as a first line) is the style tr.ClassPerso_Line_1 ", the style of the selected line style" tr.ClassPerso_Line_Select "and the style of the overflown by mouse is the style tr.ClassPerso_Line_Over.

For styles common lines between pairs and / or odd and / or selected and / or over it, you can set the style div.ClassPerso tbody ", such as the font used:




### 15.3.14 Object « objection »

The object "objection" of InterfaceDesigner is a complex object in the HTML "table". You can change the style of this "container" in creating the style table.ClassPerso.

The part to select the objection (in the two provisions: vertical or horizontal) is a complex object in edit mode and HTML object to a "select" once the script generated. It is possible to define his style in creating the style. ClassPerso_Select.

The "response to the objection" is the object HTML "textarea". You can define her style by creating the style. ClassPerso_textarea.

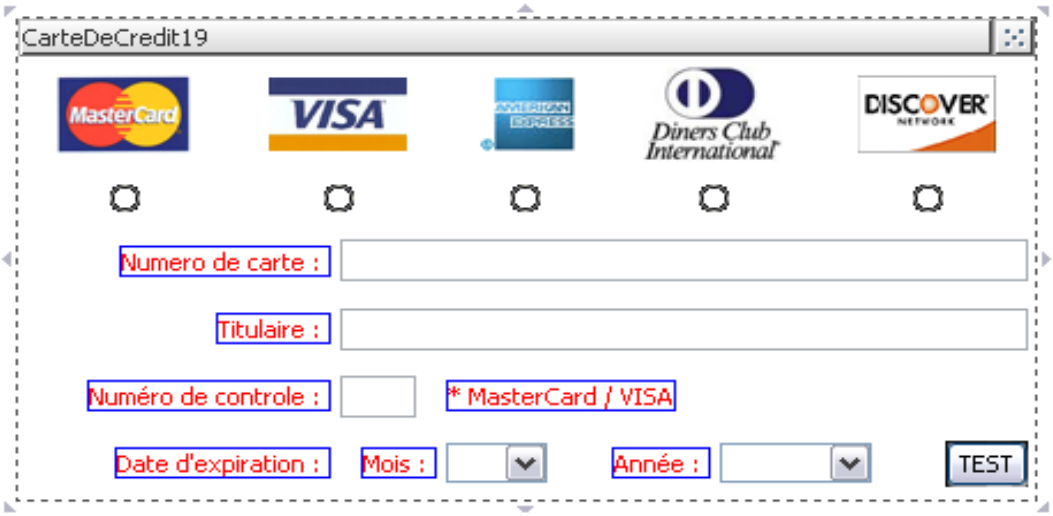
### 15.3.15 Object « IFrame »

The object "IFrame" of the generated script is the HTML "iframe". You can customize the style. However, the style will be applied in the generated script and not in InterfaceDesigner.

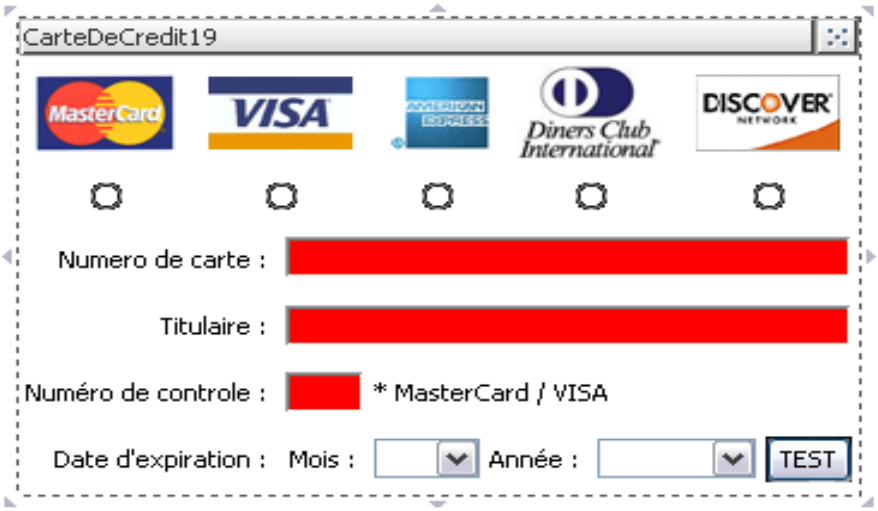
### 15.3.16 Object « credit card »

The object credit card has no property "class style", though its elements are composed of other basic elements.

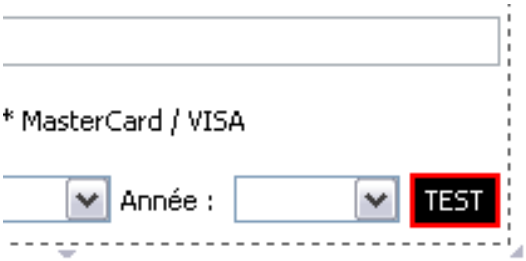
Labels use the class "span.Label_HN (default class of objects" label ") that you then customize:




The text fields using the class "input.TextBox_HN" (the default class of objects "fields") that you can customize




the "test" button uses the class "input.Button_HN" (the default class of objects "button") that you can customize:




## 15.4 Using images in the CSS files

It is possible to define background images in uploadée InterfaceDesigner to society, or the pictures by default.

The structure of InterfaceDesigner and generated scripts are not the same, the background images are visible in the generated script.

The directories "images" and "styles" in the script is generated in the same hierarchical level. So the background image should be defined as follows: background-image: url ('../ Images / Nom_IMAGE ');

Example: background-image: url ('../ Images / Logo_Vocalcom.gif ');

Moreover, it should be noted that only the images in the script are actually copied to the generated script. Indeed, the images are common to any society, and therefore it allows a reasonable generation time even when the company has a very large number of images. So if you want to set a background image in a style file and that it is not used by the script, a trick is to add an item to the script "image" with the desired image and make the 'object is not visible.

## 15.5 Using tifferent style sheets in the same script

All style files uploaded to the same script are included in the script in alphabetical order. The style file defines the script is included in the last page.
Note that if more of your style files define the same class, the definition included the last file included that is taken into account by the browser.

## 15.6 Appendix « replace database connections during script import »

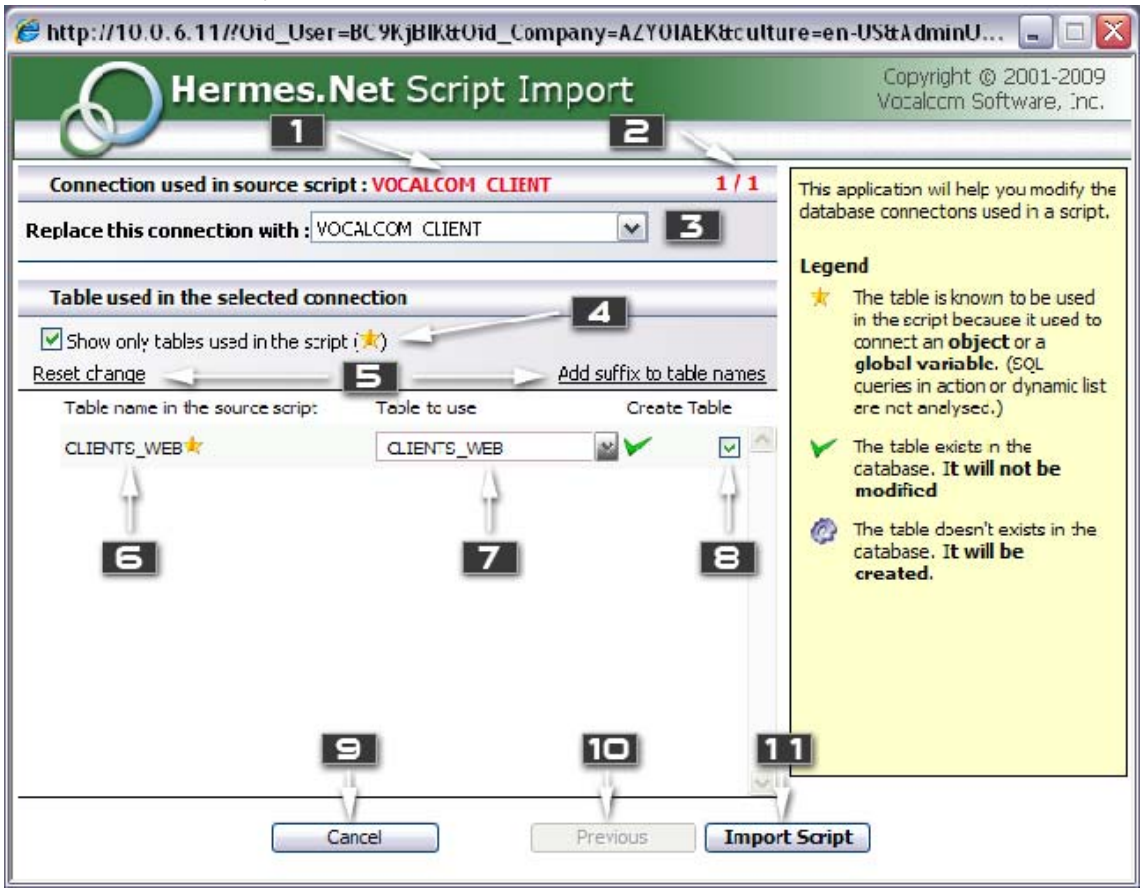
When importing a script or a duplication, you will be asked to choose how to replace the connections to existing databases.

This will allow connections for each / tables used in script to specify another table to use or create a blank table with the same structure.
It is possible to maintain existing connections.

This feature is only available on import or duplicating a script v4. If the imported script is a script v3, the screen to replace the connections will not be displayed. If the script source v4 no connection to databases, the screen will not be displayed.

Only the connections of objects and global variables can be replaced. No action will be made on the connections between the publisher of action and the connections of dynamic data objects "Data Matrix" and "Selection List".

This module is not compatible with Oracle databases.


Hermes.Net Script Import
Copyright © 2001-2009
Vocalcom Software, Inc.
1
2
Connection used in source script : VOCALCOM CLIENT
1 / 1
Replace this connection with : VOCALCOM CLIENT
3
Table used in the selected connection
4
Show only tables used in the script
Reset change
5
Add suffix to table names
Table name in the source script:
Table to use
Create Table
CLIENTS_WEB
CLIENTS_WEB
6
7
8
This application wil help you modify the database connectons used in a script.
Legend
The table is known to be used in the script because it used to connect an object or a global variable. (SQL queries in action or dynamic list are not analysed.)
The table exists in the database. It will not be modified
The table doesn't exists in the database. It will be created.
9
10
11
Cancel
Previous
Import Script

1. Connection name to replace of the source script.
Connections are replaced in turns: this screen is displayed for each one of them.
2. Displays the current step. Step one is the replacement of a connection.
3. Selection list to select the connection in the script imported. The list displays all available connections to the site.

4. This option allows you to specify whether only the tables of the connection source used by the script will be "mapped" or if all tables in the connection will be (to recover a complete database structure).
5. The button "Cancel changes" which will undo all changes of the current step.
The button "Add a suffix" to the names of tables allows to automatically add a suffix to the name of all tables requested. A prompt will ask user to choose the suffix.
6. This column displays the name of the connection tables to replace the script source.
A star replaces the table name if it is used in the script.
7. Selection list to select the table of correspondence.
This list is editable, so it is possible to enter a table name that does not exist in the new connection.
If the table does not exist, it will be displayed in red in the list and the icon will be displayed next to the list.
If the table exists, it is shown in black in the list and the icon will be displayed next to the list.
8. This check box lets you specify if the table must be created in the connection or not.
If the option is not checked the table will not be established, but connections using the objects will still be changed.
If the table exists in the new connection, it will be created or modified (you must be careful tables of the same name in two different connections, but different structure).
9. This button allows you to cancel the import or duplication of the script.
10. If replacing connection requires several steps, this button is used to return to the previous step. If the return is impossible (the current step is the first), the button is disabled.
11. This button allows you to complete the import or script duplication by validatating the script replacements of the configured connection.
If the current step is not the last, the button is replaced by a "Next" button that allows switching to the next step.


Vocalcom - 7 rue de Tilsitt – 75017 Paris - Tél. : +33 1 55 37 30 50 – Fax : +33 1 55 37 30 51 – www.vocalcom.com
SA au capital de 5 000 000 € – RCS Paris B 401 973 631 – TVA Intracommunautaire : FR 27401973631